City, University of London

School of Mathematics, Computer Science and Engineering

Developing a Cardiac Mechanics Analysis Framework using Shell Finite Elements

Yew Yan Wong

PhD Thesis
December 2019

Supervisors	Examiners
Prof. Roger Crouch	Dr. William Coombs
Dr. Shiqiang Yan	Dr. James May

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering at City, University of London,
December 2019

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my sincerest gratitude to my supervisor, Professor Roger Crouch. He always encouraged me to question the *5W1H* and to break down the problem into its simplest form. I enjoyed our discussions in the meetings so much that I actually looked forward to the meetings nearing the end of my PhD studies. I also want to thank Dr. Shiqiang Yan for exposing me to the idea of studying for a PhD as well as guiding me through my preliminary stages of numerical computing.

This would not have been possible if it wasn't for my family's unconditional love and both continued emotional and physical support throughout my life, especially my parents, Ai Chung Wong and Leh Hung Tiang.

Finally, I would also like to give thanks to Linjing Fang. It was a coincidence that we met when she took the exchange program and worked with me for my very first model of the electro-mechanical coupling model. However, I was very grateful for this chance encounter as she exposed me to a greater world of learning, her continual cheerfulness and eagerness to learn have inspired me to look at every obstacle as an opportunity to better myself. She also exposed me to a huge range of different disciplines that I never would have came across and to appreciate. It was with her guidance that we simulated my first *ionic* (FitzHugh-Nagumo) *model* through the use of electronic components to better understand the *meaning* behind the equations.

Declaration

The author grants full power of discretion to the Director of Library Services (City, University of London) to allow this thesis to be copied in whole or part without further consent from myself.

Yew Wong, December 2019

Abstract

Cardiovascular disease is the largest cause of sickness and premature death. In-vivo cardiac experimental studies are possible, but are obviously invasive. As an alternative, one could undertake a numerical simulation of a (virtual) beating heart. Current numerical simulations require an enormous amount of computational power (up to 17 hours on a 300,000 core supercomputer for a 1.5s heart beat). These simulations are usually carried out by making use of tetrahedral finite element meshes. The codes used for these simulations are invariably kept within the company that created them and this hinders the wider community from contributing to further development.

This document reports on the creation of a novel open source *Total Lagrangian* geometrically non-linear degenerated 9-noded shell-based *finite element* analysis capability (that is, a new computer code written in MATLAB) to simulate the essence of the dynamic behaviour of a beating human heart. The cardiac muscles are modelled using a simplified Torrent-Guasp geometry. The mechanical model is coupled with the Aliev-Panfilov formulation to simulate the action potential sweeping across the heart, resulting in the muscle contraction. This document provides the background to the physiology of the heart and both describes the theory and algorithms used to create the model.

The code produced demonstrated the ability to duplicate the key trends observed in cardiac behaviour. All the MATLAB scripts used in this work are available for download on `https://github.com/Yeuyo/Thesis`. This numerical model allows additional analyses and modification to be carried out by the wider community.

Abbreviations

ACh Acetylcholine

ADP Adenosine Diphosphate

ATP Adenosine Triphosphate

ATPase Adenylpyrophosphatase

CT Computerised Tomography

ECG Electrocardiogram

FE Finite Element

GP Gauss point

MRI Magnetic Resonance Imaging

ODE ordinary differential equations

PDE partial differential equations

Nomenclature

α Aliev-Panfilov ionic model constant

β Element surface area to volume ratio

ϵ Small strain

ε_a Uniaxial active strain

γ Aliev-Panfilov ionic model constant

θ Circumferential angular coordinate

θ_L Left helix angle

θ_R Right helix angle

κ Bulk modulus

λ Lamé's first parameter

μ Shear modulus

μ_1 Aliev-Panfilov ionic model constant

μ_2 Aliev-Panfilov ionic model constant

ν Poisson's ratio

ω Weighting of a *Gaussian integration point*

ϕ Rotation about $\{V_1\}$

Φ Normalised membrane potential

ψ Rotation about $\{V_2\}$

σ Cauchy stress

ξ Local coordinate

$\tilde{\xi}$ Local coordinate across the width of the myocardial band

η Local coordinate

ζ Local coordinate

$\{\varepsilon_{ini}\}$ Initial displacement effect vector (in Green strain)

$\{\varepsilon_{lin}\}$ Linear componenets of the Green strain increment vector

$\{\varepsilon_{nli}\}$ Non-linear componenets of the Green strain increment vector

$\{\sigma\}$ Cauchy stress vector

$[\varepsilon]$ Green-Lagrangian strain matrix

$[\sigma]$ Cauchy stress matrix

b Width

c Aliev-Panfilov ionic model constant

d Depth

f_b Body force

f^{Φ} Membrane potential source term

f^r Recovery variable function

h Thickness of shell element

k_{Ta} Active stress amplitude

r Recovery variable for Aliev-Panfilov ionic model

s Surface (over which integration is performed)

t Time

u Deformation along the X-axis

v Deformation along the Y-axis

w Deformation along the Z-axis

x Global deformed coordinate

y Global deformed coordinate

z Global deformed coordinate

B Aliev-Panfilov ionic model constant

C_0 Shifts ventricles *up* or *down* longitudinal axis

C_1 Stretches/squashes ventricles longitudinally

C_2 Controls band *helix angle*

C_3 Affects rate of change of *helix angle*

C_4 Controls *crossing-zone*

C_5 Controls *mid-height* location of anterior surface

C_6 Controls ventricle *compactness*

C_7 Gives Z coordinate of apex

C_m Membrane capacitance

D_{ξ} Conductivity parallel to the fibres

D_{η} Conductivity perpendicular to the fibres

D_{ex} Extracellular conductivity parallel to the fibres

D_{ey} Extracellular conductivity perpendicular to the fibres

D_{ix} Intracellular conductivity parallel to the fibres

D_{iy} Intracellular conductivity perpendicular to the fibres

E Young's modulus

F External force

I_{app} Ionic current applied externally

I_{ex} External excitation introduced

I_{ion} Ionic current generated from ionic models

I_{yy} Second moment of area about the y-axis

$\bar{J}$ Ratio of deformed volume to initial volume

J_{ion} Ionic model non-linear function

L Length

N Shape function

$\bar{N}$ Mean value of shape functions

R^{Φ} Membrane potential source residual

R^{r} Recovery variable residual

T_a Uniaxial active stress

V Initial volume

V^{AP} Membrane potential

V_d Deformed volume

V_e Extracellular potential

V_i Intracellular potential

V_m Transmembrane potential

W Heart geometry constant

X Global undeformed coordinate

Y Global undeformed coordinate

Z Global undeformed coordinate

$\{a\}$ Arbitrary nodal value vector

$\{c\}$ Set of weight functions

$\{d\}$ Displacement vector

$\{e_\xi\}$ Local ξ-direction vector

$\{e_\eta\}$ Local η-direction vector

$\{e_\zeta\}$ Local ζ-direction vector

$\{e_x\}$ Local x-direction vector

$\{e_y\}$ Local x-direction vector

$\{e_z\}$ Local z-direction vector

$\{f\}$ Force vector

$\{f_a\}$ Internal force vector (from electrophysiology)

$\{f_{int}\}$ Internal force vector

$\{\bar{f}_{int}\}$ Internal force vector for layered shell elements

$\{f_{oob}\}$ Out-of-balance force vector

$\{f_{react}\}$ Reaction force vector

$\{g_1\}$ Shell rotation displacement influence vector about $\{V_2\}$

$\{g_2\}$ Shell rotation displacement influence vector about $\{V_1\}$

$\{n\}$ Surface normal

$\{t_s\}$ Surface traction vector

$\{x\}$ Deformed coordinate vector

$\{G\}$ Shell thickness direction derivative vector

$\{S\}$ Second Piola-Kirchhoff stress vector

$\{S_a\}$ Active second Piola-Kirchhoff stress vector

$\{S_T\}$ Total second Piola-Kirchhoff stress vector

$\{V_1\}$ Shell membrane direction vector

$\{V_2\}$ Shell membrane direction vector

$\{V_n\}$ Shell thickness direction vector

$\{X\}$ Initial coordinates

$[b]$ Left Cauchy-Green matrix

$[e]$ Eulerian/Almansi strain matrix

$[k_e]$ Element stiffness matrix

$[A]$ Displacement matrix (re-organised)

$[B]$ Strain-displacement matrix

$[B^{TL}]$ Strain-displacement matrix for Total Lagrangian approach

$[B_L^{TL}]$ Linear component of Total Lagrangian strain-displacement matrix

$[B_{NL}^{TL}]$ Non-linear component of Total Lagrangian strain-displacement matrix

$[B_s^{TL}]$ Shell strain-displacement matrix for Total Lagrangian approach

$[B_{sL}^{TL}]$ Linear component of Total Lagrangian shell strain-displacement matrix

$[B_{sNL}^{TL}]$ Non-linear component of Total Lagrangian shell strain-displacement matrix

$[B^{UL}]$ Strain-displacement matrix for Updated Lagrangian approach

$[C]$ Right Cauchy-Green deformation matrix

$[D]$ Constitutive matrix

$[D_p]$ Global diffusivity matrix

$[D_{psh}]$ Local diffusivity matrix

$[F]$ Deformation gradient matrix

$[G]$ Full strain-displacement matrix for shell elements

$[H]$ Second Piola-Kirchhoff stress matrix (re-organised)

$[I]$ Identity matrix

$[J]$ Jacobian matrix

$[J_d]$ Jacobian matrix based on deformed configuration

$[K]$ Global geometrically linear stiffness matrix

$[K^L]$ Geometrically linear stiffness matrix

$[K_p]$ Equivalent *stiffness* matrix for the propagation model

$[K^{TL}]$ Total Lagrangian stiffness matrix

$[K_L^{TL}]$ Linear component of Total Lagrangian stiffness matrix

$[K_{NL}^{TL}]$ Non-linear component of Total Lagrangian stiffness matrix

$[K_s^{TL}]$ Total Lagrangian stiffness matrix for shell elements

$[\bar{K_s^{TL}}]$ Total Lagrangian stiffness matrix for layered shell elements

$[K_{sL}^{TL}]$ Linear component of Total Lagrangian stiffness matrix for shell elements

$[K_{sNL}^{TL}]$ Non-linear component of Total Lagrangian stiffness matrix for shell elements

$[K^{UL}]$ Updated Lagrangian stiffness matrix

$[K_L^{UL}]$ Linear component of Updated Lagrangian stiffness matrix

$[K_{NL}^{UL}]$ Non-linear component of Updated Lagrangian stiffness matrix

$[K_V^{UL}]$ Volume conserving component of Updated Lagrangian stiffness matrix

$[M_p]$ Equivalent of *mass* matrix for the propagation model

$[N]$ Shape function matrix

$[Q_{sh}]$ Global to local transformation matrix

$[R]$ Rotation matrix

$[S]$ Second Piola-Kirchhoff stress matrix

$[U]$ Right stretch matrix

$[V]$ Left symmetric stretch matrix

1

Introduction

1.1 Background and motivation for the study

It is well known that cardiovascular disease is the largest cause of sickness and premature death in humans (Employment, Social Policy, Health and Consumer Affairs; 2004). Among European adults, cardiovascular disease caused more than 4 million deaths (close to half of all the deaths) in 2014 (Nichols et al.; 2014). Furthermore, it is estimated that almost 5000 babies are born with congenital heart disease every year in the United Kingdom alone (British Heart Foundation; 2003). Based on the type and complexity of the defect, differing forms of treatment may be required. It is found that those who require surgical intervention to survive their infancy have reduced neuro-development outcomes, often resulting in poorer motor functions, reduced intelligence and language skills (Hovels-Gurich et al.; 2006). Anything that can be done to avoid those outcomes would clearly be beneficial. Detecting cardiovascular disease early can help the medical professions predict the progression of the disease to aid in the treatment, limiting the morbidity and mortality of the disease (Smith et al.; 2011).

In-vivo cardiac experimental studies are possible, but are obviously invasive, expensive and provide limited information about the behaviour of the heart because only a limited number of sensors can be used (Ruth; 2014). In contrast, using numerical computer-based simulations of the heart can allow a more complete study of the

behaviour, as many parametric analyses may be performed relatively easily once the initial model has been created (Plank et al.; 2009) (Kerckhoffs et al.; 2006).

As the heart in each individual differs quite significantly, it is not appropriate to simply use simulation results from an average geometry when proposing the optimal surgical intervention for a specific patient (Smith et al.; 2011). With the continuing advancement in imaging techniques, more accurate and detailed personalised cardiac models can be created (Amr et al.; 2016). This has allowed cardiac simulations to further approach the position where they will be able to enable medical teams to plan better surgical operations for the patients (Sermesant et al.; 2012), design new drugs (Yamashita et al.; 2015) and train medical staff (Aggarwal et al.; 2010) (McKinney et al.; 2013) (Allan et al.; 2010). However, at present the simulations that contain enough information to be of use for a surgical team still require an enormous amount of computational power (for example, it took up to 17 hours to simulate a 1.5s heartbeat with a 700,000 core supercomputer (Fujitsu; 2015) (TOP500; 2016)). Such computing resources are generally not available to a medical team.

This thesis first reviews the physiology of the heart and looks into the different chemo-physical processes required to undertake cardiac simulations. The thesis then presents a novel computationally efficient Finite Element analysis tool used to carry out those simulations and reveal valuable findings in relation to how the key electro-mechanical parameters control the effectiveness of the pumping action.

1.2 Components of a cardiac simulation

A meaningful cardiac simulation depends upon a significant number of parameters, including the geometry of the heart, a realistic representation of the muscle tissue mechanics, a proper modelling of the cardiac electro-chemical potential causing muscle contraction plus an appropriate simulation of the flow of blood into and out of the organ (Sugiura et al.; 2012). Each of these components necessitates detailed study. Due to the large scope of work required in a complete cardiac simulation, research groups normally focus on just one or two of those areas.

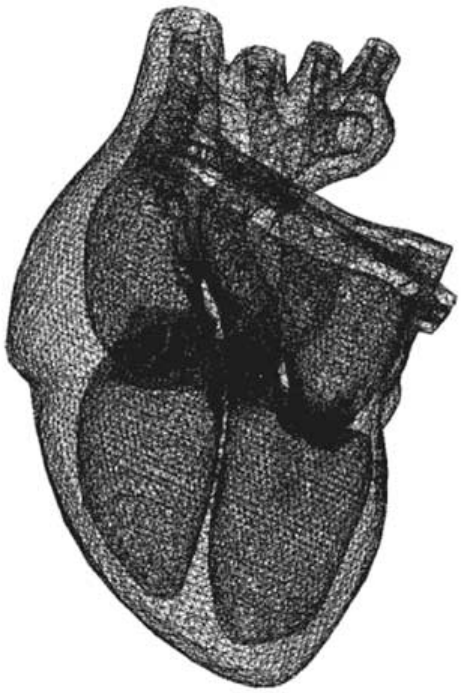


Figure 1.1: 3D wire-frame representation of a human heart (Zhang et al.; 2005)

The cardiac geometry can be obtained using an imaging technique such as computed tomography (CT) or magnetic resonance imaging (MRI) in conjunction with electrocardiogram (ECG) scans (Amr et al.; 2016) (Kayvanpour et al.; 2015). These data are then used to create a geometric model, often by employing a machine-learning approach (Zheng et al.; 2008) (Wang et al.; 2013) (Kayvanpour et al.; 2015). The 3D wire-frame geometric model (for example, see Figure 1.1) then requires a physico-chemical analysis capability to enable a prediction of the heart contraction cycle (and hence the pumping action) to be made. The physics to be simulated covers two areas: (i) the electro-chemical behaviour and (ii) the mechanical behaviour. The resulting two models can be coupled and solved simultaneously.

The electro-chemical behaviour can be expressed through use of nonlinear differential equations to duplicate a biological neuron model (Fitzhugh; 1961) (Nagumo et al.; 1962) or attempt to more directly model the ion exchanges (Hodgkin and Huxley; 1952) (Noble; 1962) (Beeler and Reuter; 1977) (Luo and Rudy; 1991) (Nash and Panfilov; 2004). These two approaches will be further elaborated in Section 5.1.

The mechanical behaviour of the heart is defined in terms of the relation between its deformation and the stresses acting within the heart tissue. That deformation is

driven by the electro-chemical behaviour, whereby an electrical pulse sweeps around the heart, resulting in the contraction of the heart muscle (see Section 2.2). The deformation-stress relationship is simulated by decomposing the stresses into *passive* and *active* components, with the passive contribution depending (via the constitutive model) solely on the state of deformation. Whereas, the active stresses are caused by the ion exchanges computed from the electro-chemical behaviour.

It is not entirely straight-forward (and it is computationally demanding) for the mechanical and electro-chemical models to be coupled fully (Kuijpers et al.; 2012). Therefore, weakly coupled approaches are commonly employed (Ruth; 2014). In these cases, the electrical potentials in the cardiac muscle are generated with the electro-chemical models, transmitted through a propagation model (see Section 5.2) and the local active stress computed from it. Such an approach is known as *one-way coupling*, as the displacement will have no effect on the electrophysiology.

When considering the geometry of the heart (which leads to the generation of a finite element mesh), it has been noted that this organ could be unfolded (after a single incision has been made (Torrent-Guasp et al.; 2005)) to reveal a twisted band with its fibre direction aligned in the long axis direction as shown in Figure 1.2. Consideration of the deformation of this continuous helical band greatly aids our understanding of the contraction and twisting responsible for the pumping action. Here we focus on the left ventricle, as this chamber is responsible for the primary action of the organ (see Chapter 2). In the work reported here, the blood flow is not modelled. While this might appear to be a significant omission, most existing cardiac simulation tools similarly have concentrated solely on the electro-mechanical response. Once realistic models for these processes exist, then the presence of the blood could be introduced through dynamic fluid-structure interaction analysis.

It is observed that the helical muscle fibers that form the left ventricle allow the heart to reach a high ejection fraction with a far lower fiber shortening compared to the case if the heart comprised muscle fibres arranged purely circumferentially. For example, reducing the helix angle (this angle is measured from the basal end of the heart's long axis to the fibre orientation) leads to a higher ejection fraction ($EF = \frac{V_d - V_s}{V_d}$, where V_d is the left ventricle volume at the end of *diastole* and V_s is the

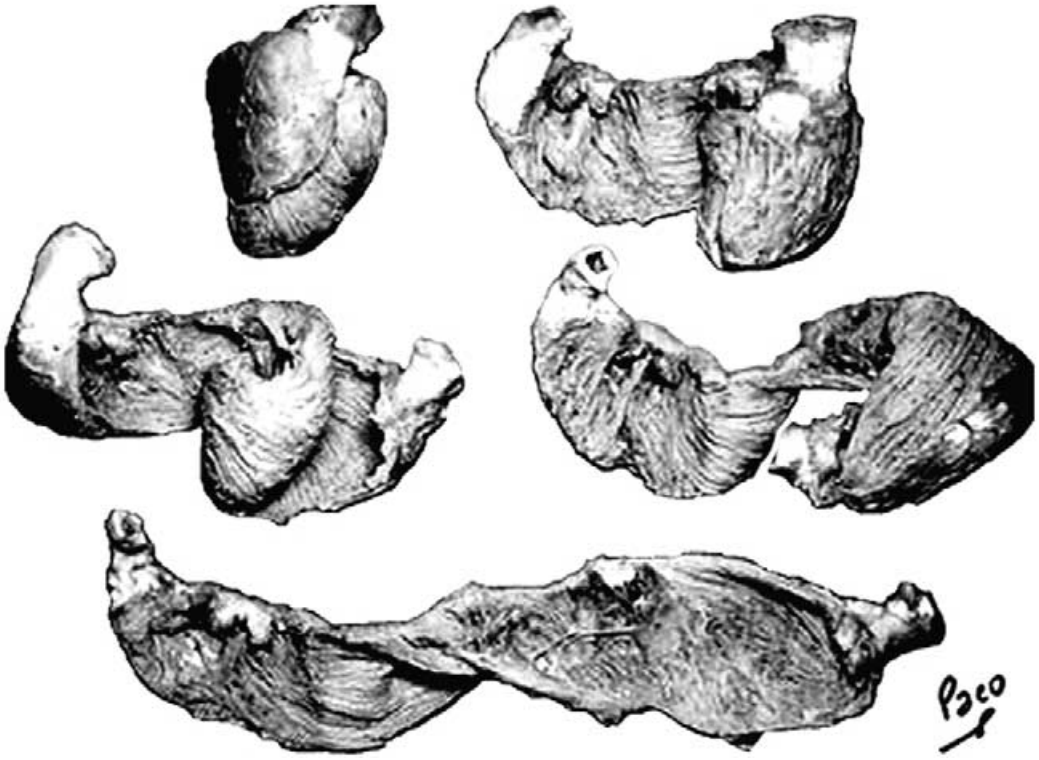


Figure 1.2: Unfolding of the heart into a single twisted band (Torrent-Guasp et al.; 2005).

left ventricle volume at the end of *systole*) (Sallin; 1969). Recent MRI imaging has confirmed the importance of the Torrent-Gausp inspired double helical band model in predicting the motion of this geometrically complex organ (Carreras et al.; 2016). This thesis does not use an ellipsoidal idealisation of the left ventricle (adopted by many other researchers) but follows the approach first employed by Grosberg (Grosberg; 2008) where a simplified Torrent-Gausp band defines the left ventricle geometry. However, unlike the work of Grosberg (which made use of hexahedral Finite Elements within the ABAQUS software) here a new coupled Finite Element code is created using shell elements in order to achieve greater computational efficiency. This code is used to simulate various different conditions to illustrate how the ejection fraction and degree of twisting are strongly influenced by the shape of the left ventricle. The analysis method has the potential to offer a useful preliminary diagnostic tool for medical practitioners.

The remainder of this thesis is broken down into the following Chapters:

In Chapter 2, the mechanisms behind the multi-scale processes occurring during

a *cardiac cycle* are reviewed. The electro-mechanical behaviour is described from the molecular level up to the contraction of the myocardium. This provides the necessary background to understand the idealisations considered in later chapters.

Chapter 3 reviews three dimensional *finite element* stress analysis,. This is the technique which will be used to simulate the behaviour of the heart. The chapter begins with the consideration of a linear solid (with 8-noded hexahedral elements). The capability of this technique is demonstrated through an example and the MATLAB script provided. The FE code is further expanded to incorporate the *geometrically non-linear* capability. This new code is tested against a established benchmark problem. Hyperelastic *material non-linearity* and *incompressibility* are also reviewed in this chapter.

Chapter 4 replaces the hexahedral elements with shell finite elements. The formulation for these elements is described in some detail with the code provided. Benchmark problems are used to compare the accuracy and computational run-time with the hexhaedral elements used in Chapter 3. Shell element layering is also considered in this chapter.

Chapter 5 looks into capturing the cardiac electrophysiology described in Chapter 2. The Aliev-Panfilov model is used to represent the membrane potential behaviour and the propagation is modelled using a Crank-Nicolson time-stepping scheme. The electro-mechanical coupling behaviour is demonstrated and the effect of mesh densities and time stepping size explored.

Chapter 6 collects the methodologies described in Chapter 4 and 5 to enable simulations of the myocardial band to be made. This chapter shows the importance of the helical shape of that band (describing by the Torrent-Guasp model) and examines the effect of different parameters on the twist and ejection fraction produced by a pumping heart. Several different heart geometries are considered in addition to an analysis where stiffer material is used to model a myocardial infarct.

References

Aggarwal, R., Mytton, O. T., Derbrew, M., Hananel, D., Heydenburg, M., Issenberg, B., MacAulay, C., Mancini, M. E., Morimoto, T., Soper, N., Ziv, A. and Reznick, R. (2010). Training and simulation for patient safety, *Quality and Safety in health Care* **19(2)**: 34–43.

Allan, C. K., Thiagarajan, R. R., Beke, D., Imprescia, A., Kappus, L. J., Garden, A., Hayes, G., Laussen, P. C., Bacha, E. and Weinstock, P. H. (2010). Simulation-based training delivered directy to the pediatric cardiac intensive care unit engenders preparedness, comofort and decreased anxiety among multidisciplinary resuscitation teams, *The Journal of Thoracic and Cardiovascular Surgery* **14(3)**: 646–652.

Amr, A., Kayvanpour, E., Sedaghat-Hamedani, F., Passerini, T., Mihalef, V., Lai, A., Neumann, D., Georgescu, B., Buss, S., Mereles, D., Zitron, E., Posch, A. E., Würstle, M., Mansi, T., Katus, H. A. and Meder, B. (2016). Personalized computer simulation of diastolic function in heart failure, *Genomics Proteomics Bioinformatics* **14**: 244–252.

Beeler, G. W. and Reuter, H. (1977). Reconstruction of the action potential of ventricular myocardial fibres, *The Journal Of Physiology* **268(1)**: 177–210.

British Heart Foundation (2003). Congenital heart disease statistics.

Carreras, F., Ballester, M., Pujadas, S., Leta, R. and Pons-Llado, G. (2016). Morphological and functional evidences of the helical heart from non-invasive cardiac imaging, *European Journal of Cardio-thoracic Surgery* **29S(S50-55)**.

Employment, Social Policy, Health and Consumer Affairs (2004). 2586th council meeting, `http://europa.eu/rapid/press-release_PRES-04-163_en.htm?locale=en`.

Fitzhugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane, *Biophysical Journal* **1(6)**: 445–466.

Fujitsu (2015). Using supercomputers for safer heart surgery, `http://www.fujitsu.com/global/vision/customerstories/university-of-tokyo/`.

Grosberg, A. (2008). *A Bio-Inspired Computational Model of Cardiac Mechanics: Pathology and Development*, PhD thesis, California Institute of Technology.

Hodgkin, A. and Huxley, A. (1952). A quantitative description of membrane current and it's application to conduction and excitation in nerve, *The Journal of Physiology* **117**: 500–544.

Hovels-Gurich, H. H., Konrad, K., Skorzenski, D., Nacken, C., Minkenberg, R., Messmer, B. J. and Seghaye, M.-C. (2006). Long-term neurodevelopmental outcome and exercise capacity after corrective surgery for tetralogy of fallot or ventricular septal defect in infancy, *The Annals Of Thoracic Surgery* **81**: 958–966.

Kayvanpour, E., Mansi, T., Sedaghat-Hamedani, F., Amr, A., Neumann, D., Georgescu, B., Seegerer, P., Kamen, A., Hass, J., Frese, K. S., Irawati, M., Wirsz, E., King, V., Buss, S., Mereles, D., Zitron, E., Keller, A., Katus, H. A., Comaniciu, D. and Meder, B. (2015). Towards personalized cardiology: Multi-scale modeling of the failing heart, *PLoS ONE* **10(7)**.

Kerckhoffs, R. C., Healy, S. N., USYK, T. P. and McCulloch, A. D. (2006). Computational methods for cardiac electromechanics, *Proceedings of the IEEE* **94**: 769–783.

Kuijpers, N. H. L., Hermeling, E., Bovendeerd, P. H. M., Delhaas, T. and Prinzen, F. W. (2012). Modeling cardiac electromechanics and mechanoelectrical coupling in dyssynchronous and failing hearts, *J. of Cardiovasc. Trans. Res.* **5**: 159–169.

Luo, C. H. and Rudy, Y. (1991). A model of the ventricular cardiac action potential. depolarization, repolarization, and their interaction, *Circulation Research* **68(6)**: 1501–1526.

McKinney, J., Cook, D. A., Wood, D. and Hatala, R. (2013). Simulation-based training for cardiac auscultation skills: Systematic review and meta-analysis, *Journal of General Internal Medicine* **28(2)**: 283–291.

Nagumo, J., Arimoto, S. and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon, *Proceedings of the IRE* **50(10)**: 2061–2070.

Nash, M. P. and Panfilov, A. V. (2004). Electromechanical model of excitable tissue to study reentrant cardiac arrhythmias, *Progress in Biophysics & Molecular Biology* **85**: 501–522.

Nichols, M., Townsend, N., Scarborough, P. and Rayner, M. (2014). Cardiovascular disease in europe 2014: Epidemiological update, *European Heart Journal* **35(42)**: 2950–2959.

Noble, D. (1962). A modification of the hodgkin - huxley equations applicable to purkinje fibre action and pacemaker potentials, *The Journal Of Physiology* **160(2)**: 317–352.

Plank, G., Burton, R. A., Hales, P., Bishop, M., Mansoori, T., Bernabeu, M. O., Garny, A., Prassal, A. J., Bollensdorff, C., Mason, F., Mahmood, F., Rodriguez, B., Grau, V., Schneider, J. E., Gavaghan, D. and Kohl, P. (2009). Generation of histo-anatomically representative models of the individual heart: tools and application, *Philosophical Transactions of The Royal Society A* **367**: 2257–2292.

Ruth, A. (2014). *Electromechanical Large Scale Computational Models of the Ventricular Myocardium*, PhD thesis, Universitat Politècnica de Catalunya.

Sallin, E. A. (1969). Fiber orientation and ejection fraction in the human left ventricle, *Biophysical Journal* **9(7)**: 954–964.

Sermesant, M., Chabiniok, R., Chinchapatnam, P., Mansi, T., Billet, F., Moireau, P., Peyrat, J., Wong, K., Relan, J., Rhode, K., Ginks, M., Lambiase, P., Delingette, H., Sorine, M., Rinaldi, C., Chapelle, D., Razavi, R. and Ayache, N. (2012). Patient-specific electromechanical models of the heart for the prediction of pacing acute effects in crt: A preliminary clinical validation, *Medical Image Analysis* **16**: 201–215.

Smith, N., de Vecchi, A., McCormick, M., Nordsletten, D., Camara, O., Frangi, A. F., Delingette, H., Sermesant, M., Relan, J., Ayache, N., Krueger, M. W., Schulze, W. H., Hose, R., Valverde, I., Beerbaum, P., Staicu, C., Siebes, M., Spaan, J., Hunter, P., Weese, J., Lehmann, H., Chapelle, D. and Rezavi, R. (2011). euheart: Personalized and integrated cardiac care using patient-specific cardiovascular modelling, *Interface Focus* **1**: 349–364.

Sugiura, S., Washio, T., Hatano, A., Okada, J., Watanabe, H. and Hisada, T. (2012). Multi-scale simulations of cardiac electrophysiology and mechanics using the university of tokyo heart simulator, *Progress in Biophysics and Molecular Biology* **110**: 380–389.

TOP500 (2016). K computer, sparc64 viiifx 2.0ghz, tofu interconnect, `https://www.top500.org/system/177232`.

Torrent-Guasp, F., Kocica, M. J., Corno, A. F., Komeda, M., Carreras-Costa, F., Flotats, A., Cosin-Aguillar, J. and Wen, H. (2005). Towards new understanding of the heart structure and function, *European Journal of Cardio-thoracic Surgery* **27**: 191–201.

Wang, Y., Georgescu, B., Chen, T., Wu, W., Wang, P., Lu, X., Ionasec, R., Zheng, Y. and Comaniciu, D. (2013). Learning-based detection and tracking in medical imaging: A probabilistic approach, *Lecture Notes in Computational Vision and Biomechanics* **7**: 209–235.

Yamashita, T., Ueda, A., Mitsui, T., Tomonaga, A., Matsumoto, S., Kodama, T. and Fujitani, H. (2015). The feasibility of an efficient drug desgin method with high-performance computers, *Chemical and Pharmaceutical Bulletin* **63**: 147–155.

Zhang, Y., Bajaj, C. B. and Sohn, B.-S. (2005). 3d finite element meshing from imaging data, *Comput Methods Appl Mech Eng.* **194**: 48–49.

Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M. and Comaniciu, D. (2008). Four-chamber heart modeling and automatic segmentation for 3d cardiac ct volumes using marginal space learning and steerable features, *IEEE Transactions on Medical Imaging* **27(11)**: 1668–1681.

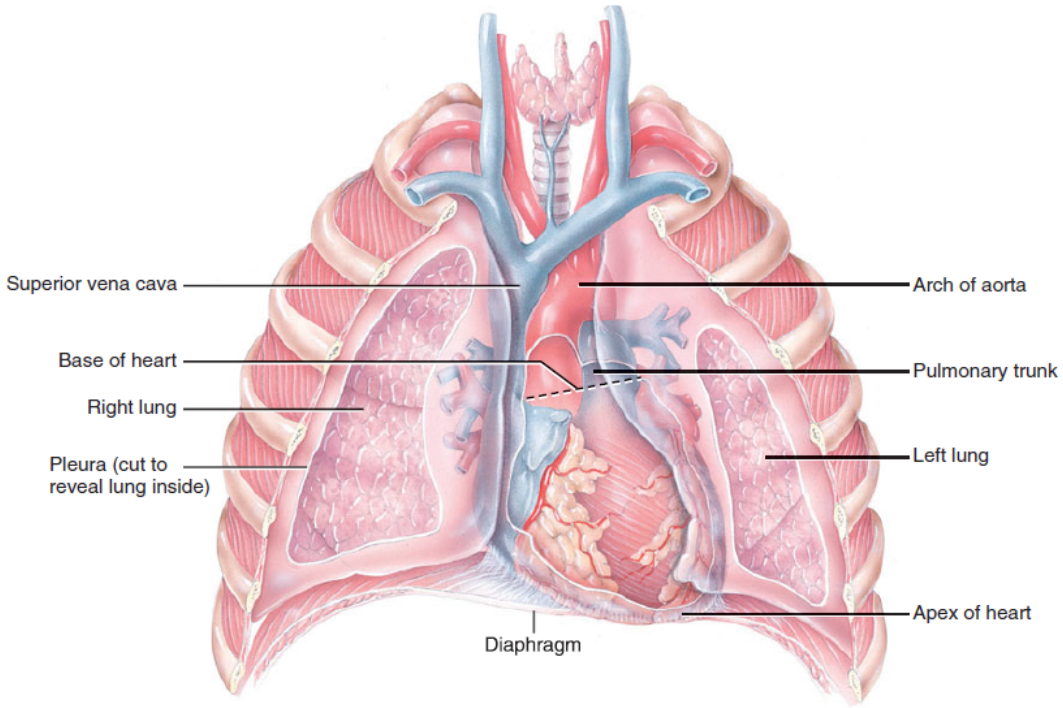
2

Physiology of the human heart

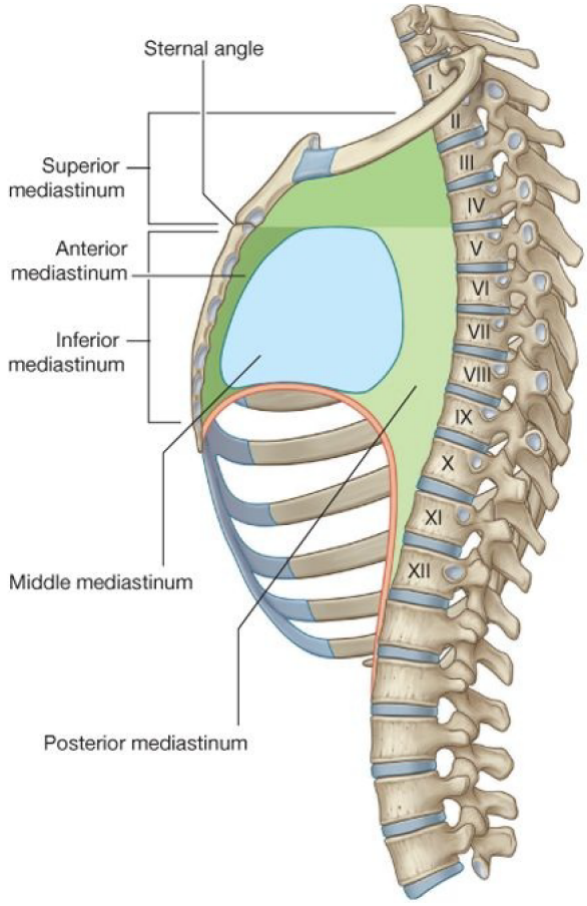
The heart is a vital organ. It allows the cells in our bodies to be able to undertake their principal functions by providing a constant supply of oxygen and nutrients through continuous blood flow. A normal healthy adult human heart is approximately 12cm long by 9cm wide. It has a mass ranging from 250-350g and pumps at an average of 70 times a minute (Levick; 2011) (Hall; 2011) (Guyton and Hall; 2011). A newborn infant heart is much smaller (approximately the size of the baby's clenched fist) and pumps at a higher average rate of 190 times a minute.

The heart is located in the *thorax* (bounded by the neck and the abdomen), between the third rib and the sixth rib as shown in Figure 2.1a. The particular part of the thorax where the heart resides is known as the *mediastinum*. The part of the *mediastinum* closer to the front of the chest is known as the *anterior mediastinum*. Behind this lies the *middle mediastinum* and then the *posterium mediastinum* (Figure 2.1b). The heart sits within the *anterior mediastinum*.

This chapter will recall and summarise the heart features and the chemistry controlling its functions. This will provide a preliminary account of the complexity that a heart simulation model is required to address.



(a) Frontal view of heart location (Jenkins et al.; 2010)



(b) Lateral view of thorax (Drake et al.; 2009)

Figure 2.1: Heart location

2.1 Heart structure and composition

The heart is a 4 chamber organ shaped like an inverted cone with two thirds of its mass on the anatomical left side[1] (as shown in Figure 2.2) . The (lowest most) conical end of the heart is known as the *apex*. The heart is held in place by a surrounding fibrous sac known as the *pericardium*. This is filled with lubricant (*pericardial fluid*, which is secreted by the heart's *serous membrane*, the *visceral pericardium*) to reduce friction as the heart contacts other internal structures while beating. The *pericardium* consists of two layers. The external layer is known as the *fibrous pericardium* and the inner layer known as the *patrietal pericardium*.

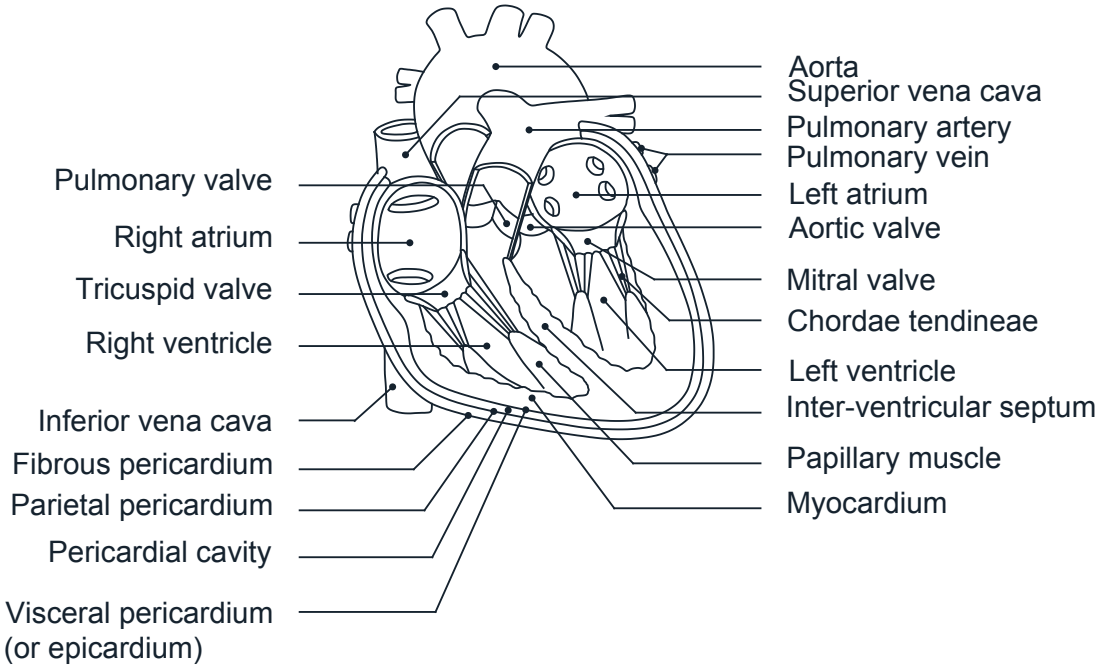


Figure 2.2: Cross-section of a human heart

The *epicardium* forms one of three layers of the heart walls (Figure 2.2). It consists of a further two layers: the *squamous cell layer* and the *areolar tissue layer*. The *squamous cell layer* is responsible for the secretion of the *pericardial fluid*. The *areolar tissue layer* is composed of loose tissue that connects the *epicardium* to the next layer of the heart. The adjacent layer is known as the *myocardium* which makes up most of the heart wall. It consists of numerous layers of cardiac muscle fibres that are responsible for the contractions in the heart (as shown in Figure 2.3).

[1]Note that the orientation of the heart referred to here is from the perspective of the host, not the viewer.

Finally, the inner-most layer (the *endocardium*) is comprised of further layers of *areolar tissue* to connect the muscle fibres of the *myocardium* to the *endocardium*. The *endocardium* covers the 4 heart valves and it supports and surrounds the blood vessels that are connected to the heart.

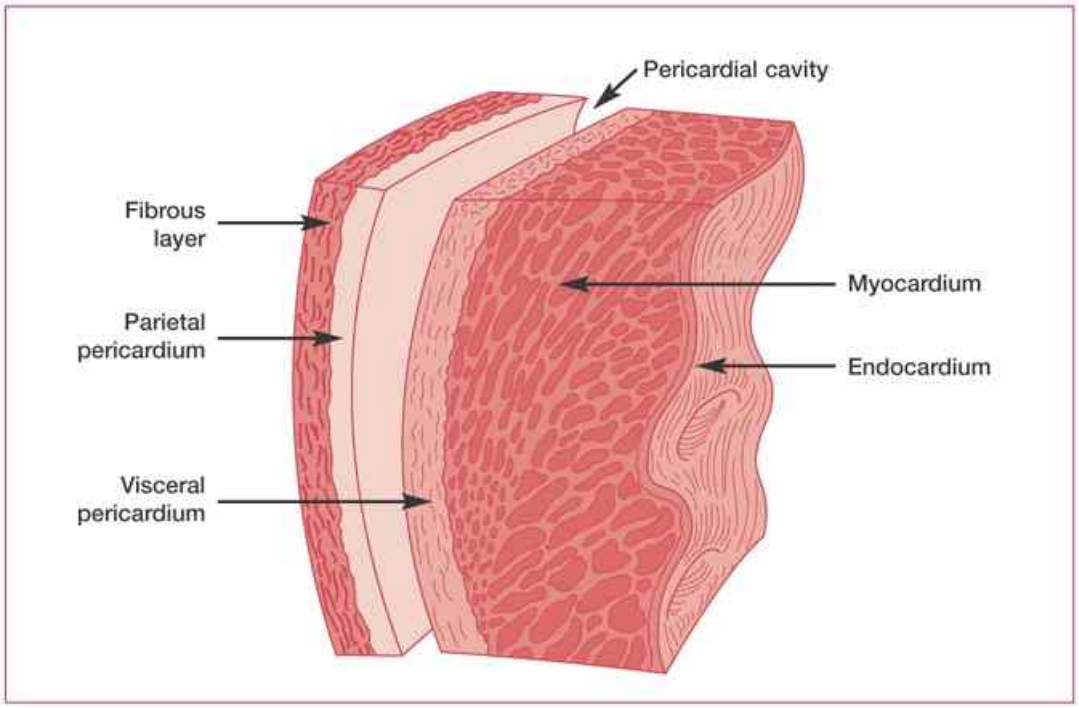


Figure 2.3: Layers of the heart wall (Dr.S.Venkatesan; 2008). Note that the total wall thickness varies from 10 to 15mm (Jenkins et al.; 2010).

The left and right sides of the heart are separated by a muscular wall known as the *interventricular septum*. Both the left and right sides of the heart are further divided into upper and lower chambers known as *atria* and *ventricles* respectively. The division between the atria and the ventricles is referred to as the *basal plane*. The *atria* are where the heart first receives blood from the body.

The vessels carrying blood from the body to the right *atrium* are the *superior* (upper) *vena cava* and *inferior* (lower) *vena cava*. Both are approximately 20mm in diameter (Prince et al.; 1983). The vessel that carries blood from the right *ventricle* to the lungs is known as the *pulmonary artery* (mean diameter of 25mm) which splits into two further arteries after exiting the heart (Truong et al.; 2012). The 4 vessels that bring the blood back to the left *atrium* from the lungs are known as *pulmonary veins* (left superior, left inferior, right superior and right inferior veins which are approximately sized at 10mm, 9mm, 12mm and 13mm diameter respec-

tively (Kim et al.; 2005)). Lastly, the vessel that carries the freshly re-oxygenated blood from the left *ventricle* to the body is known as the *aorta*, which has a diameter of approximately 33mm (decreasing to 24mm beyond the *aortic arch* (Wolak et al.; 2008)).

Atrioventricular valves separate the lower chambers from the upper chambers of the heart to prevent blood from flowing back into the upper chambers when the lower chambers are contracting. These valves are individually named; the *mitral valves* (or *bicuspid valves*) for the valves on the left side of the heart and *tricuspid valves* for those on the right side.

There are also valves between the *ventricles* and the blood vessels that carry the blood away from the heart. These are known as *semi-lunar valves*. They have the same purpose as the *atrioventricular valves*; which is to prevent back flow of blood when the *ventricles* start to relax. These valves are named according to the blood vessels they connects to. That is, the *aortic valve* and *pulmonary valve* for the left and right sides respectively.

The *atrioventricular valves* move passively. They close when blood flows backwards in the direction of the *ventricles* to the *atria*, and open when the blood pushes forward from the *atria* to flow into the *ventricles*. The shape of these valves allow them to close with very little back-flow, whereas the heavier *semi-lunar valves* require a slightly larger back-flow before they close (Guyton and Hall; 2011). The *atrioventricular valves* between the chambers are connected to the *papillary muscles* in the wall of the *ventricles* through *chordae tendineae*. The *papillary muscles* contract when the ventricle walls contract. However, they do not play any part in helping the closing of the valves. Instead, they pull the valves inwards towards the *ventricles* during the ventricular contraction to prevent the valve flaps from entering the *atria* (Guyton and Hall; 2011).

The differences in the role of the left and right sides of the heart leads to a difference in the wall thickness. The left *ventricle* has to pump blood across a greater distance (through the whole body), while the right *ventricle* only needs to pump the blood over a shorter distance (between the heart and the lungs).

The cardiac wall itself is composed of muscle fibres that are long, thin and cylindrical (ranging from 50×10^{-6} m to 100×10^{-6} m in length and 10×10^{-6} m to 20×10^{-6} m in diameter (Jenkins et al.; 2010)). They are connected both in series and parallel as shown in Figure 2.4. The muscle fibres are joined end-to-end in series by irregularly shaped sarcolemma known as *intercalated discs*. These discs consist of: (i) *desmosomes* that are responsible for attaching the cardiac cells together and (ii) *gap junctions*, which allow the electro-chemical signal (*action potentials*) in the cardiac muscle to travel along the fibres. This extra branching characteristic allows a faster spread of *action potentials* perpendicular to the longitudinal direction of the fibres. It is noted that cardiac muscle fibres can remain contracted for up to 15 times longer than skeletal muscles fibres (Guyton and Hall; 2011) (Jenkins et al.; 2010). The muscle behaviour will form the main focus of this thesis. Attempts to capture such behaviour will be further described in Section 3.2.2.

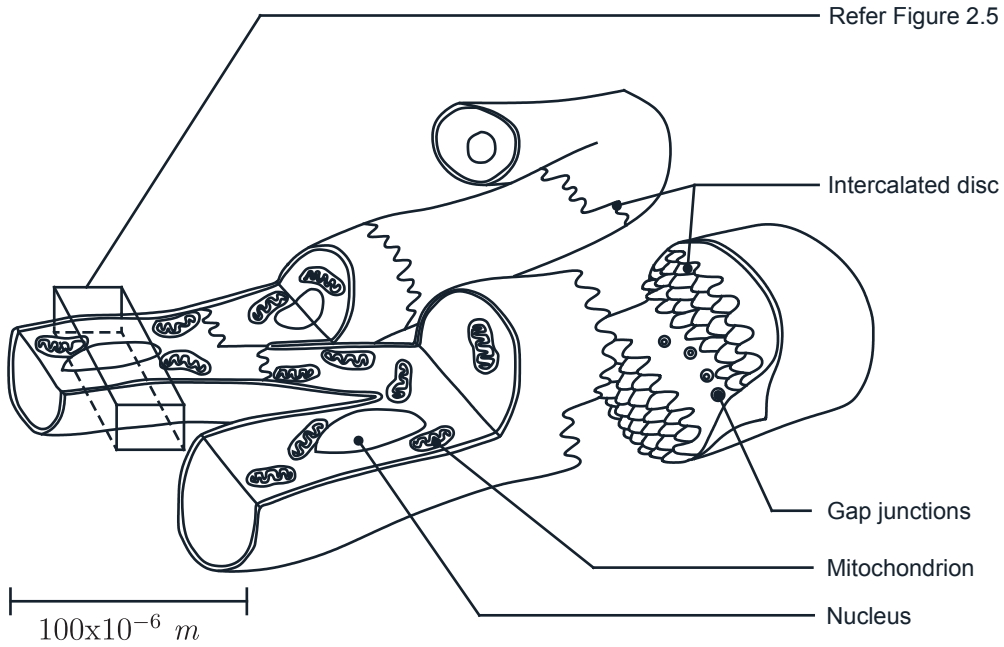


Figure 2.4: Cardiac cells

The prolonged contraction is caused by action potentials of a longer duration than seen in skeletal muscles. This occurs due to two major differences between the cardiac and skeletal muscle tissues membranes. Firstly, the action potential creating the contraction in skeletal muscles is caused by a sudden influx of sodium due to the opening of a large number of *fast sodium channels*. These channels remain open for only a few thousandths of a second (Guyton and Hall; 2011). This causes the

skeletal muscle's action potential pulse to last no more than one or two thousandths of a second. Whereas, in cardiac muscles, there are two type of channels; the same *fast sodium channels* but also *slow calcium-sodium channels*. These slow channels take longer to open, and remain open for several tenths of a second. The cardiac muscles receive their calcium ions from both the *sarcoplasmic reticulum* and from the surrounding *interstitial fluid*, whereas skeletal muscles only receive calcium ions from the *sarcoplasmic reticulum* (Jenkins et al.; 2010).

The second major difference between cardiac and skeletal muscle fibres is their permeability to potassium ions. The permeability of the cardiac muscle membrane for potassium ions decreases up to five times immediately after the start of the action potential, which is a behaviour that is not observed in normal skeletal muscles. This causes the positively charged potassium ions to remain inside the cardiac muscle cells for longer. It is only after the *calcium-sodium channels* close (0.2-0.3s after the action potential first reaches the cell) that the permeability of potassium ions increases to allow the membrane potential to return to its resting level (Guyton and Hall; 2011). In skeletal muscles, the contraction is typically caused by *acetylcholine* (ACh) released by a nerve impulse in the motor neuron. However, cardiac muscle tissue is normally stimulated by its own *autorhythmic muscle fibres* (Jenkins et al.; 2010). This difference requires the *mitochondria* in the cardiac muscle fibres to be larger and more numerous when compared with skeletal muscle fibres. The mitochondria are responsible for producing *adenosine triphosphate* (ATP). Cardiac muscle tissue is also capable of producing ATP using the lactic acid generated by skeletal muscles during heavy exercise, which helps increase the heart rate (Jenkins et al.; 2010).

A portion of the cardiac muscle tissue in the *myocardium* is illustrated in greater detail in Figure 2.5. That figure shows a short section of a cardiac fibre. The cardiac fibres are surrounded by *sarcolemma* which act as barriers between the extracellular and intracellular components. These *sarcolemma* extend into the muscle fibre with a tube-like feature known as the *transverse tubule* (*T tubule*), which extends further into the *sarcoplasmic reticulum* to aid the rapid release of calcium ions. The cardiac fibre itself is divided longitudinally into segments known as the *sarcomere* (bounded

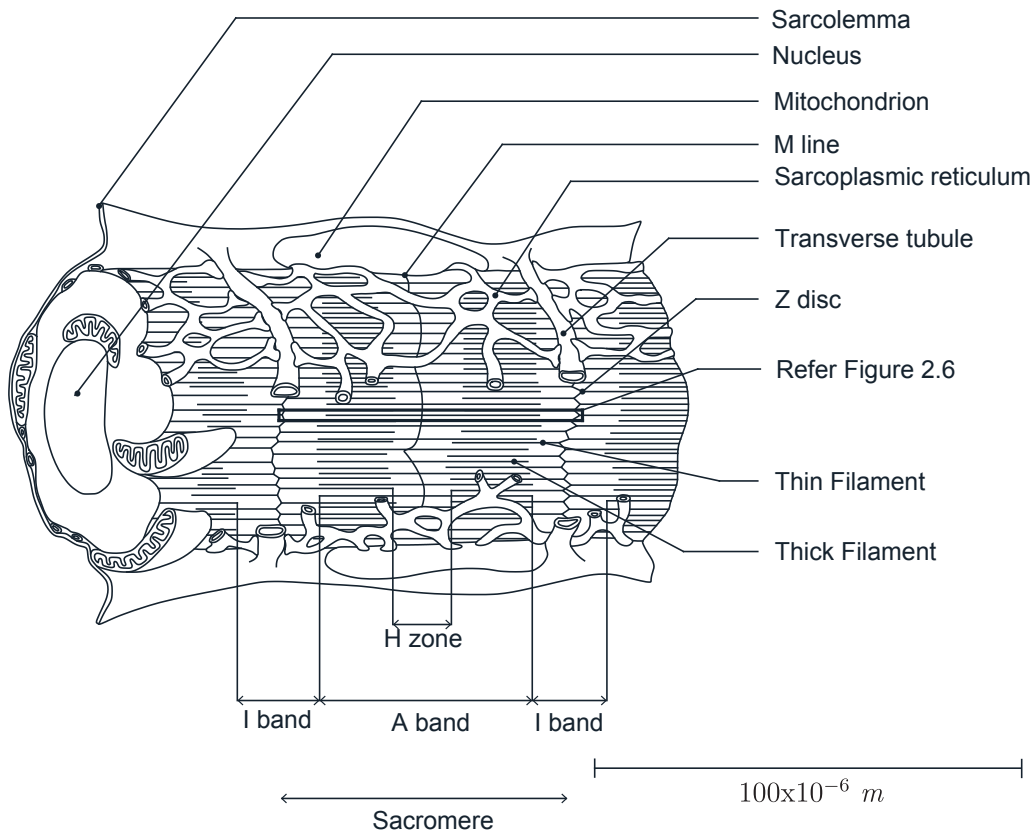


Figure 2.5: Cardiac muscle fibres details

by the *Z discs*). The distance between two *thick filaments* is known as the *I band*, of which the *Z disc* is located centrally. The *thick filament* extends to the *I band* of the next *Z disc*. The distance that the *thick filament* occupies is known as the *A band*. The *thick filament* is split into two along its centre by *M line proteins*. There is also a *thin filament* which exists between the *Z discs*. *Thin filaments* are in contact with *Z discs* but do not extend along the whole distance between the *Z discs* and have a gap in the middle where the *M line* exists. This gap between the *thin filaments* is known as the *H zone*.

A detailed (zoomed-in) view is shown in Figure 2.6. Here it can be seen that the *thick filament* is connected to the *Z discs* by the *titin* protein that acts like a spring to push the *thick filament* back to its original position after contraction.

The *thin filament*'s diameter is about 8nm and consists of mostly *G actin* forming two twisted strands (Figure 2.7) (Jenkins et al.; 2010). The two twisted strands are referred to as *F-actin*. The strand length is controlled by other proteins which span across the strand, known as *nebulin*. These are capped at the open end by

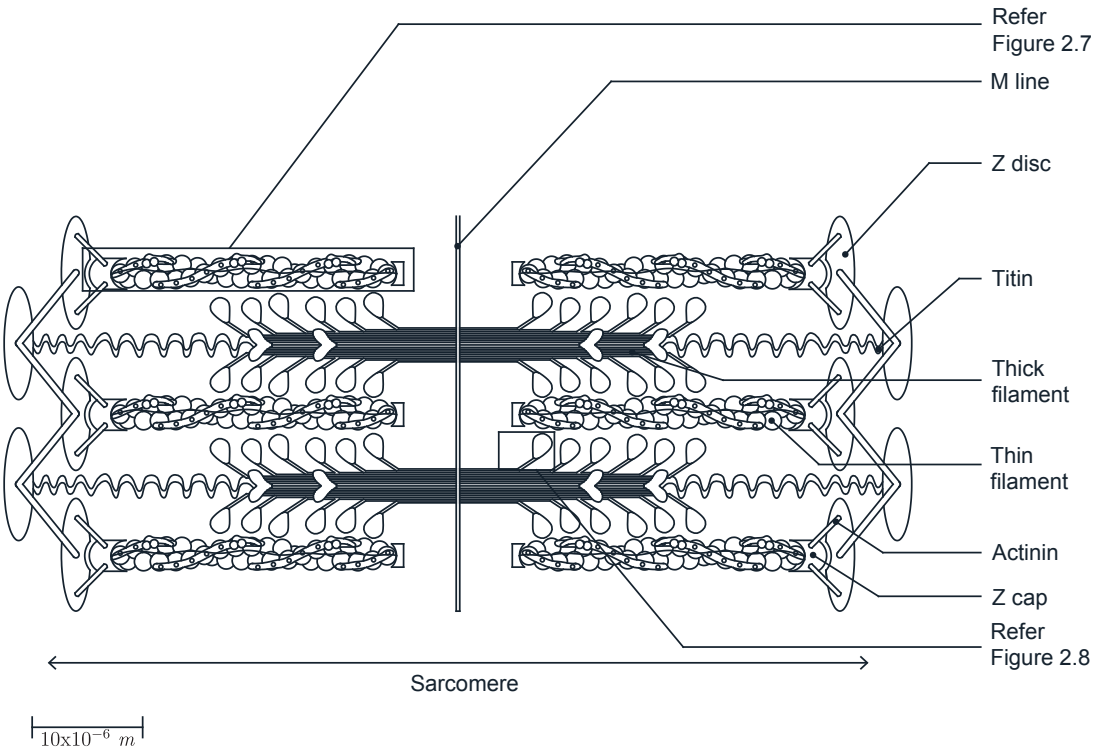


Figure 2.6: Sacromere details

tropomodulin and capped by *Cap Z* at the other end, which is connected to the *Z disc* by *actinin* proteins. On each of the *G actins*, there is a binding site for the *myosin head* to connect to. However, during resting, these binding sites are covered by further strands of proteins known as *tropomyosin*, which are regulated by *troponin*.

The *thick filaments* are about twice as thick as the *thin filament*, having a diameter of 16nm and comprise of about 300 *myosin* molecules (Jenkins et al.; 2010). These *myosin* molecules in turn are made up of two heavy chains of polypeptides which coil together. One end of the heavy chain is known as the *tail*, with the middle of the chain acting as a hinge to allow the *myosin* some degree of flexibility. On the other end of the heavy chains are two *heads*, each with some movement allowed by the untwisted part of the chain known as the *neck*. The *neck* is surrounded by four light chain polypeptides to regulate the movement of the *heads*. Each of these *heads* then consists of an *actin binding site* and an *adenylpyrophosphatase* (ATPase) site that allows the myosin to attach to the binding site on the *thin filaments* thereby enabling contractions.

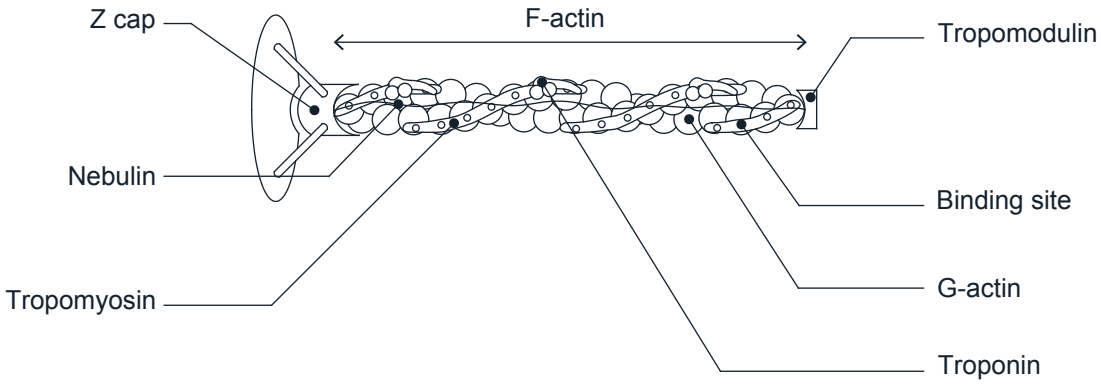


Figure 2.7: Thin filament details

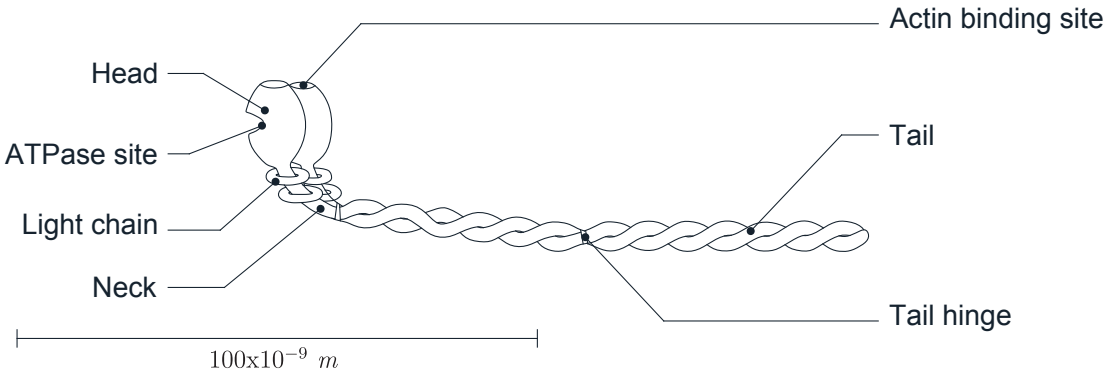


Figure 2.8: Myosin molecule

2.2 Cardiac mechanics/electro-physiology

The duration from the start of a heart beat to its end is known as the *cardiac cycle*. Each of these cycles are initiated by an action potential generated at the *sinoatrial node* which spreads around the *atrium* with the help of the *Bachmann's bundle*. The action potential's velocity in the cardiac cells is of the order of 0.3-0.5 $m.s^{-1}$. The potentials are separated between the *atrium* and *ventricle* by a fibrous tissue that surrounds the *atrioventricular valve* openings (Guyton and Hall; 2011). Thus the only path for action potentials to reach the *ventricles* is through the *atrioventricular node*. The velocity of the action potential passing through that node is about 0.05 $m.s^{-1}$ (Klabunde; 2008). This arrangement causes a delay of more than 0.1 s for the action potential to reach the *ventricles* from the *atrium* (Guyton and Hall; 2011). The action potential is then carried rapidly by the *His bundle* at a velocity of about $2m.s^{-1}$ towards the *left posterior bundle* and *right bundle* along the *interventricular*

septum at the same velocity (Klabunde; 2008). Finally, the action potential is spread through the wall of the *ventricles* by the *Purkinje fibres* at the velocity of approximately 4 $m.s^{-1}$ (Klabunde; 2008). This arrangement results in the action potential arriving at different parts of the heart at different times (see Figure 2.9) to achieve the maximum pumping efficiency possible.

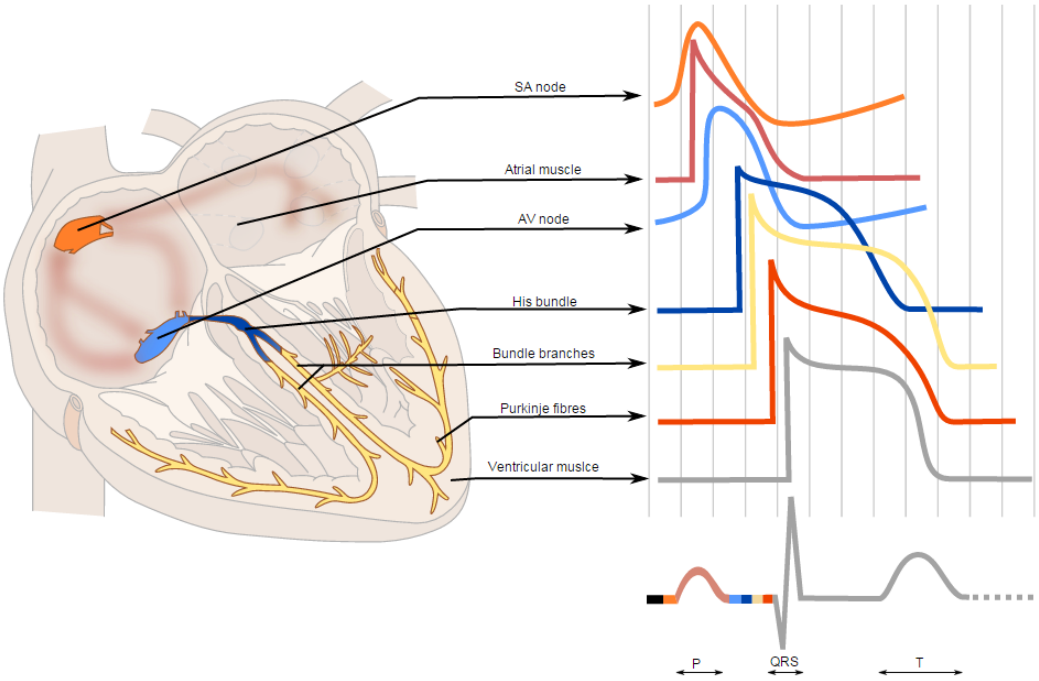


Figure 2.9: Action potential pathway and the muscles' action potential transients at different locations together with the PQRST sequence (ECG Pedia; 2011).

The complete cardiac cycle is divided into two phases, the *diastole* phase and the *systole* phase. Figure 2.9 illustrates the PQRST electrocardiogram sequence which is widely used to characterise the cardiac *signature* of an individual. During the *diastole*-phase, the blood fills the heart chambers as they relax (that is, expand). During this stage the *atrioventricular valves* open to allow the blood to flow into the *ventricles*. This enables about 80% of the blood to flow directly through the *atria* into the *ventricles*. After some time, the *atrium* contracts, pumping the remaining 20% into the ventricles. It is during the *systole* phase that the *ventricles* contract and pump blood out of the heart. At this stage the *atrioventricular valves* close to prevent blood flowing back into the *atria* from the *ventricles* and the *semi-lunar valves* open to allow the blood to flow out of the *ventricles* into their respective blood vessels. After the *ventricles* finish contracting, blood will attempt to flow back into the (now almost empty) *ventricles*, but is prevented from doing so by

the *semi-lunar valves*. This marks the end of the cardiac cycle. Note that the twisting action of the heart (where the apex rotates clockwise, when viewed from the *basal plane*, and the *basal plane* rotates anticlockwise during systole) is key to understanding the mechanics of this organ. This deformation will be explored in some detail in Chapter 6.

These cycles are caused by exchanges of ions. The process is referred to as excitation. Excitation in skeletal muscles is caused by the release of ACh (see Section 2.1) from a motor neuron to the neuromuscular junction. As the ACh reaches the synaptic sarcolemma, it opens the sodium channels on the motor end-plate (the regions where muscle tissues are connected to the nerve cells) which allows an influx of sodium into the end-plate to produce an end-plate potential.

The cardiac muscle tissue potential is increased by about 105 *millivolts*, from approximately -85 *millivolts* to $+20$ *millivolts* (Guyton and Hall; 2011). As mentioned previously, the action potential first propagates from the *sinoatrial* node. When cardiac muscle tissue is positively charged, its positivity will cause the nearby *sarcolemma* to depolarize. Once a threshold level is reached, the sodium voltage-gated channels will open to allow more sodium ions to enter the cells and cause further depolarization. When the potential difference between the *sarcolemma* converges towards the surrounding potential, the influx of sodium will slow down and the sodium channel will begin to close as potassium voltage-gated channels begin to open to allow potassium ions to diffuse out of the *sarcolemma*, which then allows for repolarization. After that, the sodium-potassium pumps will actively restore the sodium and potassium ions in the *sarcolemma* back to their original condition.

When the sarcolemma first depolarizes, it causes other gates nearby to depolarize. This starts off a chain reaction that propagates the action potential along the *sarcolemma* (illustrated as 1 in Figure 2.10), and as the channel cannot re-open until the local repolarization is complete, the action potential cannot propagate backwards to where the action potential first arose.

When the action potential reaches a *T tubule*, it then travels down the *T tubules* (illustrated as 2 in Figure 2.10), causing nearby *terminal cisternae* of the *sarcoplasmic reticulum* to open and release calcium ions into the surrounding *sarcomeres* (illus-

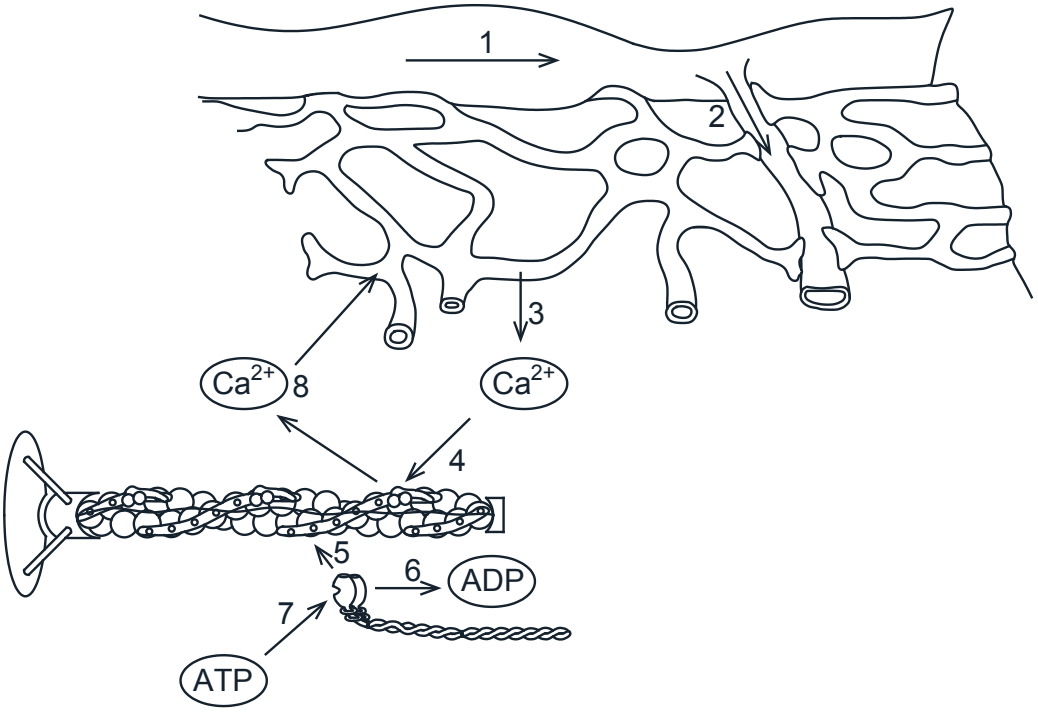


Figure 2.10: Contraction process

trated as 3 in Figure 2.10). It also noted that for cardiac muscles, the *sarcoplasmic reticulum* is less developed compared to that of skeletal muscles. Cardiac muscles have *T tubules* that are five times the diameter of the skeletal muscles, which results in a volume that is 25 times greater than that of skeletal muscles to compensate for the smaller *sarcoplasmic retiruculum* (Guyton and Hall; 2011). However, as the main source of calcium ions for cardiac muscles come from *T tubules* instead of *sarcoplasmic reticulum* inside the tissue, this means that the strength of the contraction is largely dependent on the calcium ions available in the *extracellular fluid* (Guyton and Hall; 2011). These calcium ions would have to bind to the *troponin* along the *thin filaments* and cause *tropomyosin* to move away from *myosin binding sites* (illustrated as 4 in Figure 2.10) to allow the heads of the nearby *myosin* to bind to the *G-actins* (illustrated as 5 in Figure 2.10).

As the *myosin head* binds to the *G-actins*, the *adenosine diphosphate* (ADP) molecule then leaves the *myosin head*, causing the head to lose energy and move to a new low energy position, dragging the *thin filament* towards the centre of the *sarcomere* (illustrated as 6 in Figure 2.10).

Only after new ATPs attach to the *myosin heads*, and are hydrolyzed into ADP

again, will the *myosin heads* move back into their high energy position and cause the muscle to relax once more (illustrated as 7 in Figure 2.10).

It is noted that cardiac muscle and skeletal muscle contractile forces are regulated differently. This is because skeletal muscle contraction is dependent on the recruitment and asynchronous firing of different numbers of motor units and that the individual motor unit can build up contraction through a brief series of re-excitations. Whereas cardiac muscle is normally activated simultaneously in its entire mass and the cells are electrically refractory (cannot be activated again) after the excitation, until mechanical relaxation has taken place.

The amount of calcium bound to *troponin-C* during each systole induces less than half the maximum activation of the contractile apparatus. Therefore, it is possible to produce more force by increasing the amount of calcium bound to *troponin-C*. This can be achieved by controlling the amount of free calcium that is released into the *cytosol* during systole (Gray; 2014).

A cardiac muscle action potential can last up to 0.3 s. Whereas the skeletal muscle action potential typically last just 1×10^{-3} s (Dorri; 2004). The long-lasting depolarization allows a prolonged influx of calcium, and this enables a greater amount of calcium to be stored in the *sarcoplasmic reticulum*, so that more calcium can be released in subsequent contractions. These movements allow an automatic mechanism to match increases in heart rates with an increase in contractile force (Gray; 2014).

After the contraction, the calcium ions inside the muscle tissues are pumped back into the *sarcoplasmic reticulum* by *calcium-ATPase pumps* (illustrated as 8 in Figure 2.10). Once the *sarcoplasmic reticulum* has recovered its calcium ions, the remaining calcium ions in the muscle tissues will be removed through *sodium-calcium exchanger* which depletes the calcium from the cell by taking in sodium. The sodium ions are then removed from the cell by *sodium-potassium ATPase pumps* (Guyton and Hall; 2011). These ions exchanges cycles are simulated and explained further in Chapter 5.

It can be concluded that the heart is a highly complex organ, with many components

and electro-chemical processes acting on different scales to achieve its purpose. This chapter has summarised those processes and provided an explanation of how the cardiac muscle contracts. In the following chapter an introduction to the *Finite Element Method* will be given, as this thesis will attempt to capture the heart behaviour using that numerical approach.

References

Dorri, F. (2004). *A Finite Element Model Of The Human Left Ventricular Systole, Taking Into Account The Fibre Orientation Pattern*, Swiss Federal Institute Of Technology Zurich.

Drake, R., Vogl, A. W. and Mitchell, A. (2009). *Gray's Anatomy for Students*, second edition edn, Churchill Livingstone.

Dr.S.Venkatesan (2008). What is the mechanism of pericardial rub ?, `https://drsvenkatesan.com/2008/09/28/what-is-the-mechanism-of-pericardial-rub/`.

ECG Pedia (2011). Av conduction, `http://en.ecgpedia.org/index.php?title=AV_Conduction`.

Gray, H. (2014). *Gray's Anatomy*, Churchill Livingstone Elsevier.

Guyton, A. C. and Hall, J. E. (2011). *Textbook of Medical Physiology*, 12 edn, Saunders Elsevier.

Hall, J. E. (2011). *Guyton and Hall: Textbook of Medical Physiology*, twelfth edition edn, Saunders Elsevier.

Jenkins, G. W., Kemnitz, C. P. and Tortora, G. J. (2010). *Anatomy and Physiology: From Science To Life*, 2 edn, John Wiley & Sons, Inc.

Kim, Y., Marom, E., Herndon, J. and McAdams, H. (2005). Pulmonary vein diameter, cross-sectional area and shape: Ct analysis, *Radiology* **235(1)**: 43–49.

Klabunde, R. E. (2008). Normal impulse conduction, `http://www.cvphysiology.com/Arrhythmias/A003.htm`.

Levick, J. (2011). *An Introduction To Cardiovascular Physilogy*, fifth edition edn, Hodder Arnold.

Prince, M., Novelline, R., Athanasoulis, C. and Simon, M. (1983). The diameter of the inferior vena cava and its implication for the use of vena caval filters, *Radiology* **149(3)**: 687–689.

Truong, Q. A., Massaro, J. M., Rogers, I. S., Mahabadi, A. A., Kriegel, M. F., Fox, C. S., O'Donnell, C. J. and Hoffmann, U. (2012). Reference values for normal pulmonary artery dimensions by noncontrast cardiac computed tomography, *Circulation: Cardiovascular Imaging* **5**: 147 – 154.

Wolak, A., Gransar, H., Thomson, L. E., Friedman, J. D., Hachamovitch, R., Gustein, A., Shaw, L. J., Polk, D., Wong, N. D., Saouaf, R., Hayes, S. W., Rozanski, A., Slomka, P. J., Germano, G. and Berman, D. S. (2008). Aortic size assessment by noncontrast cardiac computed tomography: Normal limits by age, gender and body surface area, *The Journal of the American College of Cardiology* **1(2)**: 200–209.

3

3D continuum finite element analysis

A central concept within continuum mechanics is that the material can be infinitely sub-divided into infinitesimally sized volumes (or points) with no reference to any structural inhomogeneities which may appear at an atomic or sub-atomic level. This dictates that the response of a continuum is described by the behaviour of an infinite number of points. A great many of the physical processes of interest to engineering (for example stress analysis or heat transfer) can be expressed mathematically using ordinary or partial differential equations (ODEs or PDEs) which describe the response at any such point. However, most of these differential equations cannot be solved analytically except for the simplest of geometries and boundary conditions. That is, the ODEs or PDEs do not, in general, possess closed-form solutions. The *finite element* (FE) method is an attractive, powerful, numerical method which may be used to overcome that limitation by approximating the solution to those differential equations.

The conventional FE method requires (i) the geometry to be partitioned into a finite number of discrete elements connected to each other at their nodes and (ii) assumptions made with regard to how the unknown variable is distributed within the element (for example, linearly or quadratically). This process of dividing the geometry into elements is known as meshing, with the entire set of elements being

known as the mesh.

The particular element chosen, and the number of nodes associated with that element, decides the total number of unknowns (referred to as the degrees of freedom) to be solved for in a problem. Thus, the FE method considers a finite number of degrees of freedom compared to an infinite number of degrees of freedom existing within a classical continuum mechanics representation. The restriction in the number of degrees of freedom and the assumption that the unknowns vary according to pre-defined *shape functions* (that is, how the unknown variable is distributed) means that in a linear elastic mechanics problem (where the user is typically interested in predicting the displacements and internal stresses given forces applied to a structure) the FE solution will always under-predict the displacements. This is a consequence of the shape functions invariably (except for the simplest of problems) over-constraining the true deformation of the structure, thereby providing a stiffer-than-true solution. As the number of finite elements increases, so the FE solution will converge to the exact solution for a linear problem if the element numbers are increased in a sensible way[1].

In this chapter the FE method is further described and its capabilities explored. The FE method will be used in later chapters to attempt to capture the complex 3 dimensional deformation of the heart during a single heartbeat, taking consideration idealisations of many of the features described in the previous chapter.

3.1 Linear analysis

The majority of engineering stress analysis problems do not involve the structure undergoing significant displacements. For those problems, the structural geometry remains essentially unchanged during loading. Furthermore, in many cases the material (or stress-strain) behaviour remains within the *linear elastic* range. These problems are classified as *linear* problems. In the following subsections, the proce-

[1]That is, increasing the number of elements in a direction (or region) where there is no change in the deformation field will not improve the accuracy of the simulation. Yet failing to have a sufficient number of elements where the deformation field varies non-linearly can lead to significant inaccuracies.

dures and key expressions used for a *linear* FE stress analysis are given. In a later section, this work will be extended to consider non-linear effects, which are essential to the study of a beating heart.

3.1.1 Coordinate systems

In the FE analysis, the individual elements need not have their edges aligned parallel to the *global coordinate system*. The use of a *local coordinate system* attached to each element (illustrated in Figure 3.1) helps greatly in the construction of the key expressions. We therefore make use of two coordinate systems

- *Global Coordinates* (X, Y, Z). The global coordinates are used to define the nodal coordinates, typically within an *orthogonal Cartesian framework*.
- *Local Coordinates* (ξ, η, ζ)[2]. The local coordinates are used to define the space within the element, with each local coordinate typically ranging from -1 to 1 from one face of the element to its opposite face.

3.1.2 Shape functions

Shape functions (N_i in (3.1)) express the assumed (typically polynomial) variation in the unknown across the element based on the values of the unknown at the nodes and the location of the point of interest. The contributions from each node for an 8-noded hexahedral element are given by (3.1), with the nodal number linked to that contribution illustrated in Figure 3.1.

$$\begin{aligned} N_1 &= \tfrac{1}{8}(1-\xi)(1-\eta)(1-\zeta) & N_5 &= \tfrac{1}{8}(1-\xi)(1+\eta)(1-\zeta) \\ N_2 &= \tfrac{1}{8}(1-\xi)(1-\eta)(1+\zeta) & N_6 &= \tfrac{1}{8}(1-\xi)(1+\eta)(1+\zeta) \\ N_3 &= \tfrac{1}{8}(1+\xi)(1-\eta)(1+\zeta) & N_7 &= \tfrac{1}{8}(1+\xi)(1+\eta)(1+\zeta) \\ N_4 &= \tfrac{1}{8}(1+\xi)(1-\eta)(1-\zeta) & N_8 &= \tfrac{1}{8}(1+\xi)(1+\eta)(1-\zeta) \end{aligned} \tag{3.1}$$

[2]Some researchers refer to the *local coordinates* as the *natural coordinates*.

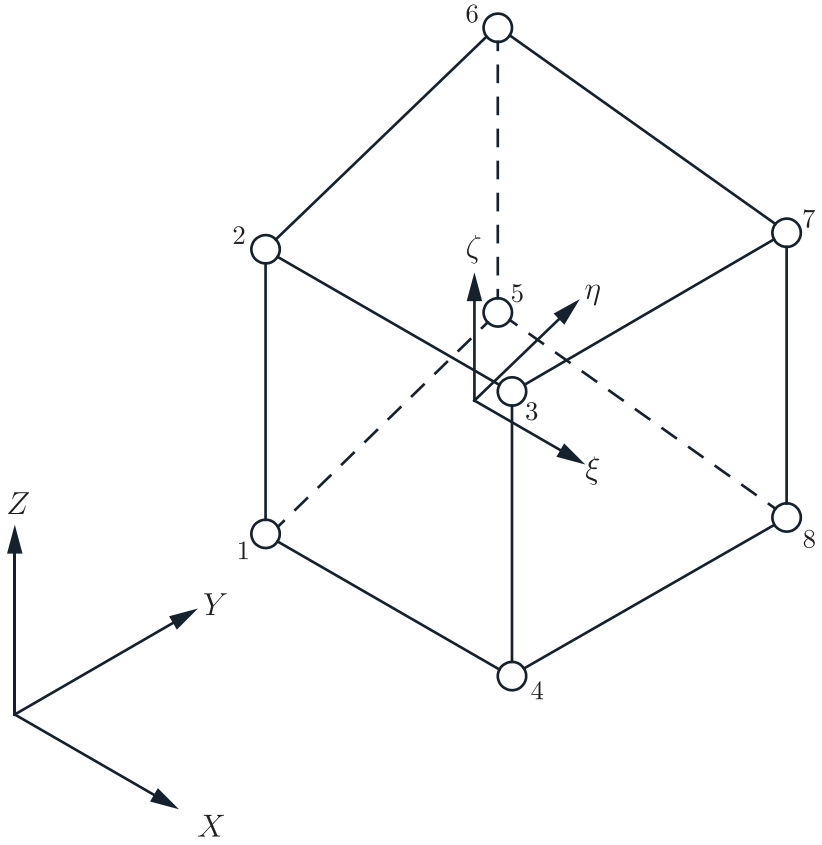


Figure 3.1: An irregular hexahedral finite element showing the global and local coordinate systems and the local node (8, in this case) numbering scheme.

3.1.3 Linear [B] (strain-displacement) matrix

An important step in the construction of the *linear* FE equations involves the use of the strain-displacement transformation matrix (hereafter referred to as $[B]$) which expresses the 6 unique (in a 3D analysis) strain components in terms of the gradient of the displacements, $\{d\}= \{u,\, v,\, w\}^T$. The 3D small strain components are given as

$$\begin{Bmatrix} \epsilon_{XX} \\ \epsilon_{YY} \\ \epsilon_{ZZ} \\ \epsilon_{XY} \\ \epsilon_{YZ} \\ \epsilon_{XZ} \end{Bmatrix} = \Big[\mathcal{L}\Big] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{3.2}$$

where the differential operator

$$\left[\mathcal{L}\right] = \begin{bmatrix} \frac{\partial}{\partial X} & 0 & 0 \\ 0 & \frac{\partial}{\partial Y} & 0 \\ 0 & 0 & \frac{\partial}{\partial Z} \\ \frac{\partial}{\partial Y} & \frac{\partial}{\partial X} & 0 \\ 0 & \frac{\partial}{\partial Z} & \frac{\partial}{\partial Y} \\ \frac{\partial}{\partial Z} & 0 & \frac{\partial}{\partial X} \end{bmatrix} \tag{3.3}$$

In order to express the strain at any location within an element, the displacements in (3.2) are computed through use of the *shape functions* (3.1) and the nodal values as follows

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \dots & N_8 & 0 & 0 \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \dots & 0 & N_8 & 0 \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \dots & 0 & 0 & N_8 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ \vdots \\ u_8 \\ v_8 \\ w_8 \end{Bmatrix} \tag{3.4}$$

where $u_1, v_1, w_1 \dots u_8, v_8, w_8$ give the X, Y and Z displacements of nodes 1 to 8. Introducing (3.4) into (3.2) gives

$$\begin{Bmatrix} \epsilon_{XX} \\ \epsilon_{YY} \\ \epsilon_{ZZ} \\ \gamma_{XY} \\ \gamma_{YZ} \\ \gamma_{XZ} \end{Bmatrix} = \Big[B\Big] \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ \vdots \\ u_8 \\ v_8 \\ w_8 \end{Bmatrix} \tag{3.5}$$

where

$$\Big[B\Big] = \begin{bmatrix} \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & 0 & 0 & \dots & \frac{\partial N_8}{\partial X} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & 0 & \dots & 0 & \frac{\partial N_8}{\partial Y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & \dots & 0 & 0 & \frac{\partial N_8}{\partial Z} \\ \frac{\partial N_1}{\partial Y} & \frac{\partial N_1}{\partial X} & 0 & \frac{\partial N_2}{\partial Y} & \frac{\partial N_2}{\partial X} & 0 & \dots & \frac{\partial N_8}{\partial Y} & \frac{\partial N_8}{\partial X} & 0 \\ 0 & \frac{\partial N_1}{\partial Z} & \frac{\partial N_1}{\partial Y} & 0 & \frac{\partial N_2}{\partial Z} & \frac{\partial N_2}{\partial Y} & \dots & 0 & \frac{\partial N_8}{\partial Z} & \frac{\partial N_8}{\partial Y} \\ \frac{\partial N_1}{\partial Z} & 0 & \frac{\partial N_1}{\partial X} & \frac{\partial N_2}{\partial Z} & 0 & \frac{\partial N_2}{\partial X} & \dots & \frac{\partial N_8}{\partial Z} & 0 & \frac{\partial N_8}{\partial X} \end{bmatrix} \tag{3.6}$$

Recognising that the nodal values are independent of the location within an element (ξ, η, ζ), the differentiation of $\{d\}$ just involves the differentiation of the *shape functions* with respect to the global coordinates. Through the use of chain rule, we determine these derivatives as follows

$$\begin{Bmatrix} \frac{\partial N_i}{\partial X} \\ \frac{\partial N_i}{\partial Y} \\ \frac{\partial N_i}{\partial Z} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial X} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial X} + \frac{\partial N_i}{\partial \zeta}\frac{\partial \zeta}{\partial X} \\ \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial Y} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial Y} + \frac{\partial N_i}{\partial \zeta}\frac{\partial \zeta}{\partial Y} \\ \frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial Z} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial Z} + \frac{\partial N_i}{\partial \zeta}\frac{\partial \zeta}{\partial Z} \end{Bmatrix} \tag{3.7}$$

The $\frac{\partial N_i}{\partial \xi}$, $\frac{\partial N_i}{\partial \eta}$ and $\frac{\partial N_i}{\partial \zeta}$ are given below for the 8-noded hexahedral element

$$
\begin{array}{lll}
\frac{\partial N_1}{\partial \xi} = -\frac{1}{8}(1-\eta)(1-\zeta) & \frac{\partial N_1}{\partial \eta} = -\frac{1}{8}(1-\xi)(1-\zeta) & \frac{\partial N_1}{\partial \zeta} = -\frac{1}{8}(1-\xi)(1-\eta) \\
\frac{\partial N_2}{\partial \xi} = -\frac{1}{8}(1-\eta)(1+\zeta) & \frac{\partial N_2}{\partial \eta} = -\frac{1}{8}(1-\xi)(1+\zeta) & \frac{\partial N_2}{\partial \zeta} = \frac{1}{8}(1-\xi)(1-\eta) \\
\frac{\partial N_3}{\partial \xi} = \frac{1}{8}(1-\eta)(1+\zeta) & \frac{\partial N_3}{\partial \eta} = -\frac{1}{8}(1+\xi)(1+\zeta) & \frac{\partial N_3}{\partial \zeta} = \frac{1}{8}(1+\xi)(1-\eta) \\
\frac{\partial N_4}{\partial \xi} = \frac{1}{8}(1-\eta)(1-\zeta) & \frac{\partial N_4}{\partial \eta} = -\frac{1}{8}(1+\xi)(1-\zeta) & \frac{\partial N_4}{\partial \zeta} = -\frac{1}{8}(1+\xi)(1-\eta) \\
\frac{\partial N_5}{\partial \xi} = -\frac{1}{8}(1+\eta)(1-\zeta) & \frac{\partial N_5}{\partial \eta} = \frac{1}{8}(1-\xi)(1-\zeta) & \frac{\partial N_5}{\partial \zeta} = -\frac{1}{8}(1-\xi)(1+\eta) \\
\frac{\partial N_6}{\partial \xi} = -\frac{1}{8}(1+\eta)(1+\zeta) & \frac{\partial N_6}{\partial \eta} = \frac{1}{8}(1-\xi)(1+\zeta) & \frac{\partial N_6}{\partial \zeta} = \frac{1}{8}(1-\xi)(1+\eta) \\
\frac{\partial N_7}{\partial \xi} = \frac{1}{8}(1+\eta)(1+\zeta) & \frac{\partial N_7}{\partial \eta} = \frac{1}{8}(1+\xi)(1+\zeta) & \frac{\partial N_7}{\partial \zeta} = \frac{1}{8}(1+\xi)(1+\eta) \\
\frac{\partial N_8}{\partial \xi} = \frac{1}{8}(1+\eta)(1-\zeta) & \frac{\partial N_8}{\partial \eta} = \frac{1}{8}(1+\xi)(1-\zeta) & \frac{\partial N_8}{\partial \zeta} = -\frac{1}{8}(1+\xi)(1+\eta)
\end{array}
\tag{3.8}
$$

The remaining components in (3.7) are obtained by determining the inverse of the derivative of the *global coordinates* with respect to the *local coordinates* as

$$
\begin{aligned}
\frac{\partial \xi}{\partial X} &= \left(\frac{\partial X}{\partial \xi}\right)^{-1} \\
&= \left(\left\{ \frac{\partial N_1}{\partial \xi} \quad \frac{\partial N_2}{\partial \xi} \quad \frac{\partial N_3}{\partial \xi} \quad \frac{\partial N_4}{\partial \xi} \quad \frac{\partial N_5}{\partial \xi} \quad \frac{\partial N_6}{\partial \xi} \quad \frac{\partial N_7}{\partial \xi} \quad \frac{\partial N_8}{\partial \xi} \right\} \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \end{Bmatrix} \right)^{-1}
\end{aligned}
\tag{3.9}
$$

where X_1 to X_8 give the X-coordinates of nodes 1 to 8. In multi-dimension case, this is done by inversing the Jacobian matrix, $[J]$ which is described as

$$
\Big[J\Big] = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \cdots & \frac{\partial N_8}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \cdots & \frac{\partial N_8}{\partial \eta} \\ \frac{\partial N_1}{\partial \zeta} & \frac{\partial N_2}{\partial \zeta} & \cdots & \frac{\partial N_8}{\partial \zeta} \end{bmatrix} \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_8 & Y_8 & Z_8 \end{bmatrix}
\tag{3.10}
$$

$$\Big[J\Big]^{-1} = \begin{bmatrix} \frac{\partial X}{\partial \xi} & \frac{\partial Y}{\partial \xi} & \frac{\partial Z}{\partial \xi} \\ \frac{\partial X}{\partial \eta} & \frac{\partial Y}{\partial \eta} & \frac{\partial Z}{\partial \eta} \\ \frac{\partial X}{\partial \zeta} & \frac{\partial Y}{\partial \zeta} & \frac{\partial Z}{\partial \zeta} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{\partial \xi}{\partial X} & \frac{\partial \xi}{\partial Y} & \frac{\partial \xi}{\partial Z} \\ \frac{\partial \eta}{\partial X} & \frac{\partial \eta}{\partial Y} & \frac{\partial \eta}{\partial Z} \\ \frac{\partial \zeta}{\partial X} & \frac{\partial \zeta}{\partial Y} & \frac{\partial \zeta}{\partial Z} \end{bmatrix} \tag{3.11}$$

3.1.4 Material properties

Undertaking *linear elastic* stress analysis using the FE method allows a variety of material behaviours to be modelled. For example, *isotropic* or *anisotropic* material responses may be considered. An isotropic material model would be expressed through the following $[D]$ constitutive matrix which links the stress components to the strain components, $\{\sigma\} = [D]\{\epsilon\}$,

$$\Big[D\Big] = \frac{E}{2(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-2\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\nu) \end{bmatrix} \tag{3.12}$$

where E is *Young's modulus* and ν is *Poisson's ratio*. Note that the arrangement within $[D]$ corresponds to the configuration of $[B]$ shown in (3.6). For a transversely isotropic material, one would need to modify the (and introduce further) elastic components related to the particular material directions. In such cases one would need to ensure that the axes of symmetry were appropriately transformed with respect to the local coordinate axes.

In the next section, both $[B]$ and $[D]$ will be used within the governing differential

equation for a general linear elastic stress analysis problem.

3.1.5 Weighted residual approach

A problem solved by the FE method seeks the values of the unknowns that satisfy the governing differential equation. The differential equations that describes the particular problem and the associated boundary conditions are collectively known as the *strong form*. Considering the static equilibrium of an infinitesimal cube, the forces acting along the X direction may be expressed by the stress variation within that cube, as shown in Figure 3.2.

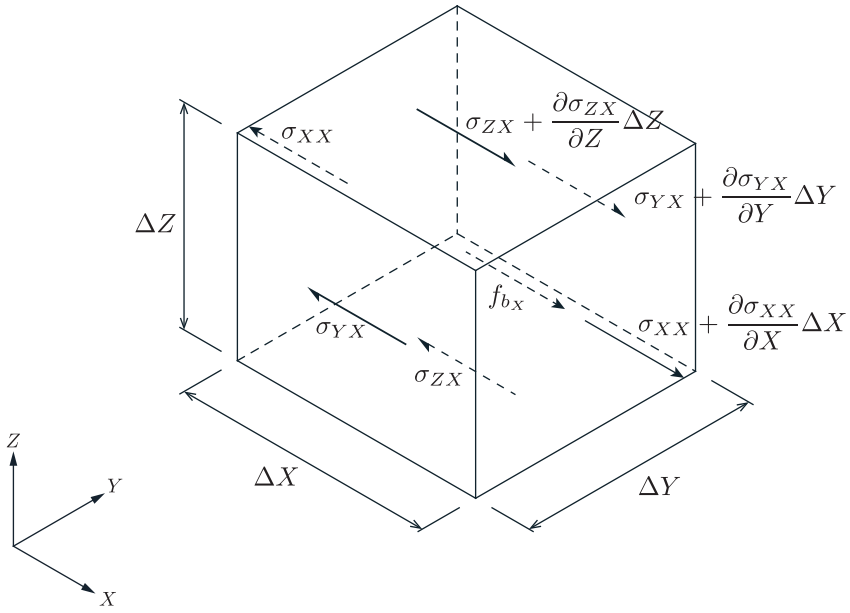


Figure 3.2: Stresses acting along the X-direction on a cube

Converting each stress component to a force (by multiplication of the area of the face on which it is acting) gives the following equation for force equilibrium in the X-direction

$$\begin{aligned}
&(\sigma_{XX} + \frac{\partial \sigma_{XX}}{\partial X}\Delta X)\Delta Y\Delta Z - \sigma_{XX}\Delta Y\Delta Z + \\
&(\sigma_{YX} + \frac{\partial \sigma_{YX}}{\partial Y}\Delta Y)\Delta X\Delta Z - \sigma_{YX}\Delta X\Delta Z + \\
&(\sigma_{ZX} + \frac{\partial \sigma_{ZX}}{\partial Z}\Delta Z)\Delta X\Delta Y - \sigma_{ZX}\Delta X\Delta X + f_{b_X}\Delta X\Delta Y\Delta Z = 0
\end{aligned} \tag{3.13}$$

where f_{b_X} are the body forces (per unit volume) acting in the X-direction. Terms cancel in (3.13) and after division by the volume and the consideration of equilibrium in the other 2 directions, we obtain

$$\begin{aligned} \frac{\partial \sigma_{XX}}{\partial X} + \frac{\partial \sigma_{YX}}{\partial Y} + \frac{\partial \sigma_{ZX}}{\partial Z} + f_{b_X} &= 0 \\ \frac{\partial \sigma_{XY}}{\partial X} + \frac{\partial \sigma_{YY}}{\partial Y} + \frac{\partial \sigma_{ZY}}{\partial Z} + f_{b_Y} &= 0 \\ \frac{\partial \sigma_{XZ}}{\partial X} + \frac{\partial \sigma_{YZ}}{\partial Y} + \frac{\partial \sigma_{ZZ}}{\partial Z} + f_{b_Z} &= 0 \end{aligned} \tag{3.14}$$

This could be written in matrix form as

$$\begin{bmatrix} \frac{\partial}{\partial X} & 0 & 0 & \frac{\partial}{\partial Y} & 0 & \frac{\partial}{\partial Z} \\ 0 & \frac{\partial}{\partial Y} & 0 & \frac{\partial}{\partial X} & \frac{\partial}{\partial Z} & 0 \\ 0 & 0 & \frac{\partial}{\partial Z} & 0 & \frac{\partial}{\partial Y} & \frac{\partial}{\partial X} \end{bmatrix} \begin{Bmatrix} \sigma_{XX} \\ \sigma_{YY} \\ \sigma_{ZZ} \\ \sigma_{XY} \\ \sigma_{YZ} \\ \sigma_{ZX} \end{Bmatrix} + \begin{Bmatrix} f_{b_X} \\ f_{b_Y} \\ f_{b_Z} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.15}$$

which may be represented as

$$\overbrace{\Big[\mathcal{L}\Big]^T}^{3\times 6} \overbrace{\Big\{\sigma\Big\}}^{6\times 1} + \overbrace{\Big\{f_b\Big\}}^{3\times 1} = \overbrace{\Big\{0\Big\}}^{3\times 1} \tag{3.16}$$

Knowing that $\{\sigma\}= [D]\{\epsilon\}$ (see Section 3.1.4), we have

$$\Big[\mathcal{L}\Big]^T \Big[D\Big] \Big\{\epsilon\Big\} + \Big\{f_b\Big\} = \Big\{0\Big\} \tag{3.17}$$

It is possible to express (3.6) using (3.3) as

$$\Big[B\Big] = \Big[\mathcal{L}\Big] \Big[N\Big] \tag{3.18}$$

Substituting (3.18) into (3.5) gives

$$\left\{\varepsilon\right\} = \left[\mathcal{L}\right]\left[N\right]\left\{d\right\} \tag{3.19}$$

Inserting (3.19) into (3.17) produces the following expression

$$\left[\mathcal{L}\right]^T\left[D\right]\left[\mathcal{L}\right]\left[N\right]\left\{d\right\} + \left\{f_b\right\} = \left\{0\right\} \tag{3.20}$$

This is the *strong form* of the governing differential equation for linear elastic force equilibrium. Finding solutions for $\{d\}$ which satisfy the *strong form* everywhere places unnecessary constraints on the continuity requirements for derivatives across element boundaries. This can be avoided by constructing what is known as the *weak form*. A more detailed explanation for this transformation may be found in numerous texts (for example, see Section 4.4.1 of (Ottosen and Petersson; 1992)).

To build the *weak form*, we multiply (3.20) by a set of *weight functions*, $\{c\}$ and integrate the expression over the finite element volume, V to obtain

$$\int_V \left\{c\right\}^T\left[\mathcal{L}\right]^T\left[D\right]\left[\mathcal{L}\right]\left[N\right]\left\{d\right\}dV + \int_V \left\{c\right\}^T\left\{f_b\right\}dV = \int_V \left\{c\right\}^T\left\{0\right\}dV = 0 \tag{3.21}$$

Applying Gauss-Green theorem (a generalisation of integration-by-parts) to the first term of (3.21),

$$\begin{aligned}\int_V \left\{c\right\}^T\left[\mathcal{L}\right]^T\left[D\right]\left[\mathcal{L}\right]\left[N\right]\left\{d\right\}dV = &-\int_V \left(\left[\mathcal{L}\right]\left\{c\right\}\right)^T\left[D\right]\left[\mathcal{L}\right]\left[N\right]\left\{d\right\}dV \\ &+ \int_s \left\{c\right\}^T\left[\sigma\right]\left\{n\right\}ds\end{aligned} \tag{3.22}$$

This operation results in the first differential operator now acting on $\{c\}$ in the volume integral. Furthermore, a surface integral (over s) now appears involving the boundary stresses $[\sigma]$ and surface normals $\{n\}$. Note that the stress terms appearing in the matrix are arranged as shown in (3.23) rather than the vector form, as shown

in (3.15).

$$\Big[\sigma\Big] = \begin{bmatrix} \sigma_{XX} & \sigma_{XY} & \sigma_{XZ} \\ \sigma_{YX} & \sigma_{YY} & \sigma_{YZ} \\ \sigma_{ZX} & \sigma_{ZY} & \sigma_{ZZ} \end{bmatrix} \tag{3.23}$$

Substituting (3.22) back into (3.21),

$$\int_s \Big\{c\Big\}^T \Big[\sigma\Big] \Big\{n\Big\}\, ds - \int_V \Big(\Big[\mathcal{L}\Big] \Big\{c\Big\}\Big)^T \Big[D\Big] \Big[\mathcal{L}\Big] \Big[N\Big] \Big\{d\Big\}\, dV + \int_V \Big\{c\Big\}^T \Big\{f_b\Big\}\, dV = 0 \tag{3.24}$$

The above expression is known as the *Galerkin* form of the weighted residual method if the *weight function*, $\{c\}$ is expressed in terms of the shape functions; that is if $\{c\} = [N]\{a\}$, where $\{a\}$ are arbitrary nodal values. Thus

$$\Big\{\Big[\mathcal{L}\Big] \Big\{c\Big\}\Big\}^T = \Big\{\Big[\mathcal{L}\Big] \Big[N\Big] \Big\{a\Big\}\Big\}^T = \Big\{a\Big\}^T \Big\{\Big[\mathcal{L}\Big] \Big[N\Big]\Big\}^T \tag{3.25}$$

Substituting (3.25) into (3.24)

$$\begin{aligned} \int_s \Big\{a\Big\}^T \Big[N\Big] \Big\{t_s\Big\}\, ds - \int_V \Big\{a\Big\}^T \Big[\Big[\mathcal{L}\Big] \Big[N\Big]\Big]^T \Big[D\Big] \Big[\mathcal{L}\Big] \Big[N\Big] \Big\{d\Big\}\, dV \\ + \int_V \Big\{a\Big\}^T \Big[N\Big]^T \Big\{f_b\Big\}\, dV = 0 \end{aligned} \tag{3.26}$$

where $\{t_s\} = [\sigma]\{n\}$ are the *surface tractions*.

Since $\{a\}$ comprises constant nodal values, they may be moved outside the integrand and subsequently removed from each side of the equation to arrive at

$$\int_s \Big[N\Big] \Big\{t_s\Big\}\, ds - \int_V \Big(\Big[\mathcal{L}\Big] \Big[N\Big]\Big)^T \Big[D\Big] \Big[\mathcal{L}\Big] \Big[N\Big] \Big\{d\Big\}\, dV + \int_V \Big[N\Big]^T \Big\{f_b\Big\}\, dV = \Big\{0\Big\} \tag{3.27}$$

Rearranging (3.27), we obtain

$$\int_V \Big(\Big[\mathcal{L}\Big] \Big[N\Big] \Big)^T \Big[D\Big] \Big[\mathcal{L}\Big] \Big[N\Big] \Big\{d\Big\}\, dV = \int_s \Big[N\Big] \Big\{t_s\Big\}\, ds + \int_V \Big[N\Big]^T \Big\{f_b\Big\}\, dV \quad (3.28)$$

Substituting (3.18) into (3.28),

$$\int_V \overbrace{\Big[B\Big]^T}^{3n\times 6} \overbrace{\Big[D\Big]}^{6\times 6} \overbrace{\Big[B\Big]}^{6\times 3n} \overbrace{\Big\{d\Big\}}^{3n\times 1} dV = \int_s \overbrace{\Big[N\Big]}^{3\times 3n} \overbrace{\Big\{t_s\Big\}}^{3n\times 1} ds + \int_V \overbrace{\Big[N\Big]^T}^{3n\times 3} \overbrace{\Big\{f_b\Big\}}^{3\times 1} dV \quad (3.29)$$

In the following subsections, the procedure to solve (3.29) numerically (that is, integrating and solving for $\{d\}$) will be explained.

3.1.6 Numerical integration

The integrals appearing in (3.29) can not be easily determined for general geometries, thus we make use of numerical integration to overcome that difficulty. As is common with most FE approaches, the *Gaussian quadrature* scheme is employed here to undertake that integration.

In the *Gaussian quadrature* scheme, a pre-defined number of sampling points are used to estimate the integrand. The number of sampling points used determines their position within the element. In this chapter, a $2 \times 2 \times 2$ *Gaussian rule* will be used. Their positions and weightings are given in Table 3.1.

Position	Weighting
$-\frac{1}{\sqrt{3}}$	1
$\frac{1}{\sqrt{3}}$	1

Table 3.1: $2 \times 2 \times 2$ Gaussian point locations and their weightings (applying to all 3 local coordinates ξ, η, ζ)

3.1.7 Stiffness matrix

The left hand side of the expression (3.29) relates the internal forces acting within the element to the displacement that the element experiences. This relationship encompasses the geometrically linear element stiffness matrix, $[K^L]$.

$$\begin{aligned}\left[K^L\right] &= \int_\Omega \left[B\right]^T \left[D\right] \left[B\right] d\Omega \\ &= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \left[B\right]^T \left[D\right] \left[B\right] d\xi d\eta d\zeta \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \left[B\right]^T \left[D\right] \left[B\right] \omega_i\omega_j\omega_k det\left[J\right]\end{aligned} \tag{3.30}$$

where ω are the weights in the *Gaussian rule*, i, j and k refer to the *Gaussian points* in each local direction and $det[J]$ is the determinant of the Jacobian.

Thus $[B]$, $[D]$ and $det[J]$ in (3.30) need to be computed at the location of each of the *Gaussian integration points*.

3.1.8 Boundary conditions

To solve the expression (3.29) (that is, to find the unknown $\{d\}$) we first need to introduce the boundary conditions. There are two common options; known as the *Dirichlet* and *Neumann* conditions. In a stress analysis problem, the former refers to the displacement being known at certain (nodal) locations whereas the later refers to a known external force (or stress) at another location.

3.1.9 Solving the linear system

The governing equation (3.29) can be simplified and expressed as $[K]\{d\} = \{f\}$ where $[K]$ and $\{f\}$ are known and the task becomes one of solving for the nodal displacements $\{d\}$. These are typically found using some variation of *Gaussian Elimination* (or an iterative process). In the codes used throughout this thesis, we make use of MATLAB m-script for all of the algorithms. The solution process

described above is achieved by the MATLAB backslash operator (as can be seen in Line 37 in Listing 3.1).

3.1.10 Example of a linear finite element mechanics algorithm

Listing 3.1 gives an example of a linear 8-noded hexahedral FE algorithm, with the procedure expressed in Figure 3.3. The number in top right of each procedure box relates to the line number in the listing. Note that the *Gauss points* (GP) calculation loops through two locations in each of the local ξ, η and ζ directions.

```
1  [E,v,coord,etopol,bc,f]=cantilever_endload;
2  nodes=size(coord,1); nDoF=3*nodes; nels=size(etopol,1);
3  eDoF=zeros(24,1); DoF=reshape(1:nDoF,3,nodes)'; K=zeros(nDoF);
4  uvw=zeros(nDoF,1); Gpsig=[]; Iz=zeros(6); Iz(1:3,1:3)=ones(3);
5  I2=eye(6); I2(4:6,4:6)=0.5*eye(3); wp=ones(8,1); dN=zeros(24,8);
6  D=E/((1+v)*(1-2*v))*(v*Iz+((1-2*v)*I2));
7  Gpcoord=zeros(nels,8,3); BeGp=zeros(nels,8,6,24);
8  xsi=[-1 -1  1  1 -1 -1  1  1]'/sqrt(3);
9  eta=[-1 -1 -1 -1  1  1  1  1]'/sqrt(3);
10 zet=[-1  1  1 -1 -1  1  1 -1]'/sqrt(3);
11 for n=1:8; sxsi=sign(xsi(n)); seta=sign(eta(n)); szet=sign(zet(n));
12   for Gp=1:8
13     N(Gp,n)=(1+sxsi*xsi(Gp))*(1+seta*eta(Gp))*(1+szet*zet(Gp));
14     dN((3*Gp)-2,n)=sxsi*(1+seta*eta(Gp))*(1+szet*zet(Gp));
15     dN((3*Gp)-1,n)=seta*(1+sxsi*xsi(Gp))*(1+szet*zet(Gp));
16     dN((3*Gp)-0,n)=szet*(1+sxsi*xsi(Gp))*(1+seta*eta(Gp));
17   end
18 end; N=N/8; dN=dN/8;
19 for nel=1:nels
20   for n=1:8
21     eDoF((3*n)-2:3*n)=DoF(etopol(nel,n),:);
22   end
23   JT=dN*coord(etopol(nel,:),:); ke=zeros(24);
24   for i=1:8; indx=[(3*i)-2:3*i]';
25     Gpcoord(nel,i,1:3)=N(i,:)*coord(etopol(nel,:),1:3);
26     detJ=det(JT(indx,:)); dNx=(JT(indx,:))\dN(indx,:);
27     B=zeros(6,24); B(1,1:3:end)=dNx(1,:); B(2,2:3:end)=dNx(2,:);
28     B(3,3:3:end)=dNx(3,:); B(4,1:3:end)=dNx(2,:);
29     B(4,2:3:end)=dNx(1,:); B(5,2:3:end)=dNx(3,:);
30     B(5,3:3:end)=dNx(2,:); B(6,1:3:end)=dNx(3,:);
31     B(6,3:3:end)=dNx(1,:); BeGp(nel,i,:,:)=B;
32     ke=ke+(B'*D*B*detJ*wp(i));
33   end
34   K(eDoF,eDoF)=K(eDoF,eDoF)+ke;
35 end
36 fDoF=[1:nDoF]'; pDoF=bc(:,1); d(pDoF)=bc(:,2); fDoF(pDoF)=[];
37 uvw(fDoF)=K(fDoF,fDoF)\(f(fDoF)-K(fDoF,pDoF)*d(pDoF));
```

Listing 3.1: 37 line MATLAB script for a 3D FE linear stress analysis using 8-noded hexahedral elements.

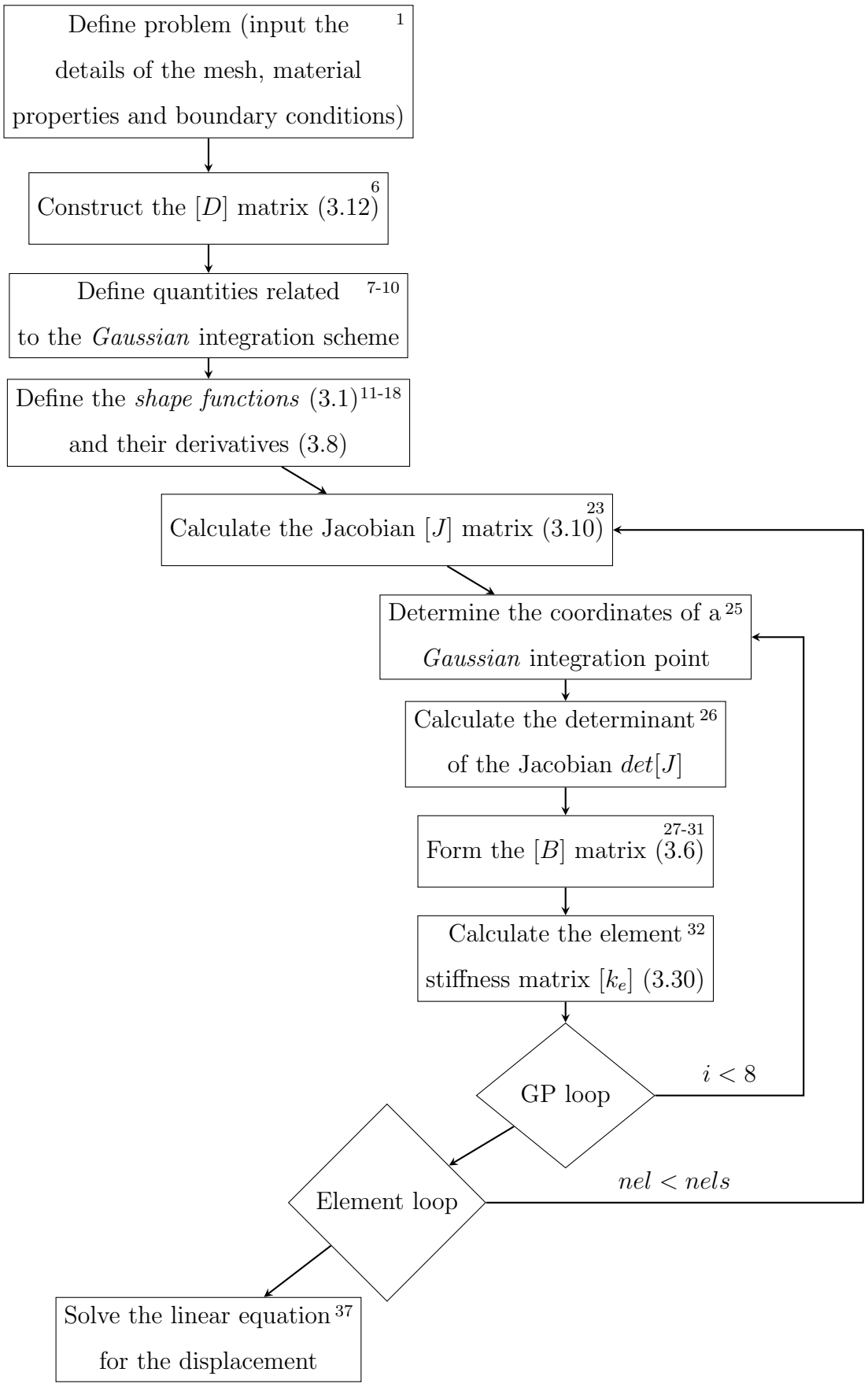


Figure 3.3: Flow diagram for the 37 line MATLAB script for a 3D FE linear stress analysis using 8-noded hexahedral elements. Note that at the end of each element loop, the stiffness contributions are added into the global stiffness matrix (line 32).

The inputs required for Listing 3.1 are summarised in Table 3.2 and the standard output shown in Table 3.3.

Variable	Matrix Sizing	Format
`E`	Scalar	E (in Pa)
`v`	Scalar	v
`coord`	$nen \times 3$	$\begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}$ (in m)
`etopol`	$nels \times nen$	$\begin{bmatrix} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 & n_7 & n_8 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$
`bc`	$nbc \times 2$	$\begin{bmatrix} node\_number & node\_displacement \\ \vdots & \vdots \end{bmatrix}$
`f`	$(nen \times dof) \times 1$	$\begin{Bmatrix} f_1 \\ f_2 \\ \vdots \end{Bmatrix}$ (in N)

Table 3.2: Input format for the linear 8-noded hexahedral analysis.

Variable	Matrix Sizing	Format
`uvw`	$(nen \times dof) \times 1$	$\begin{Bmatrix} u_1 \\ v_1 \\ \vdots \end{Bmatrix}$ (in metres)

Table 3.3: Output from the linear 8-noded hexahedral analysis.

3.1.11 Linear FE numerical example

In this section, the deflection of a simple horizontal cantilever beam is considered using the 8-noded hexahedral finite element code described earlier. The key parameters for this problem are given in Table 3.4.

Length, L	1 m
Width, b	0.1 m
Depth, d	0.1 m
Young's modulus, E	1 GPa
Poisson's ratio, ν	0
Force applied, F	1 kN

Table 3.4: Input parameters for the 8-noded hexahedral cantilever beam problem.

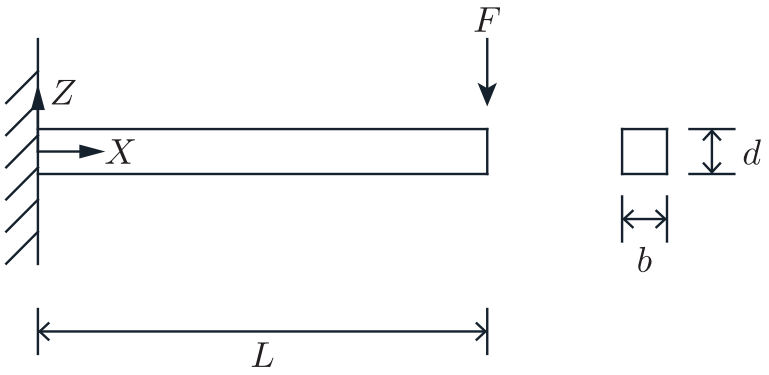


Figure 3.4: A cantilever beam subjected to end point load.

A 1 kN downwards vertical force is applied at the free end of the horizontal cantilever. The input file for this benchmark case (generating 50 8-noded hexahedral elements along its length) is given in Listing II.1.

The analytical solution for the free end displacement (w) of this beam problem (using Euler-Bernoulli theory) is given by

$$\begin{aligned} w &= -\frac{FL^3}{3EI_{yy}} \\ &= -\frac{FL^3}{3E\frac{bd^3}{12}} \\ &= -\frac{1000 \times 1^3}{3 \times 10^9 \times \frac{0.1 \times 0.1^3}{12}} \\ &= -0.0400m \end{aligned} \tag{3.31}$$

where F is the applied load and I_{yy} is the second moment of area for this rectangular prismatic beam. The simulation was run with 10, 20, 30, 40 and 50 elements along its length, yielding the results shown in Table 3.5. It can be seen that with an increase in the number of elements, the results approach the analytical solution. Euler-Bernoulli theory predicts that the displacement variation along the beam is given by a third order polynomial, yet the displacement variation along the edge of a single 8-noded hexahedral element is linear. Thus an increase in the number of elements along the length of the beam is required (or higher order shape functions used) in order to better approximate the third order polynomial. This example gives a satisfactory confirmation of the validity of this simple FE mechanics code.

Number of elements along the length	Z-displacement at the cantilever end
10	-0.0268 m
20	-0.0357 m
30	-0.0381 m
40	-0.0390 m
50	-0.0394 m

Table 3.5: End displacements resulting from the 8-noded hexahedral cantilever beam analysis.

3.2 Non-linear analysis

As mentioned previously, in many engineering applications, mechanical analyses are performed under the assumption that the material behaviour remains *linearly elastic* and the strains are treated as being infinitesimal. Furthermore, small displacements and rotations are assumed to occur (that is, the change in the structural geometry as a consequence of the loading is too small to influence the structural stiffness). However, there are cases where these assumptions are no longer valid.

3.2.1 Geometric non-linearity

The cardiac muscle fibres (within the myocardium) can shorten by up to 20% during a heartbeat. This contraction, coupled with the cone-like shape of the chambers of the heart (and the spiral orientation of the fibres) causes the organ (particularly the left ventricle) to undergo large changes in its geometry (Grosberg; 2008). Therefore, a linear analysis is inappropriate and one must follow a *geometrically non-linear* analysis.

Geometrically non-linear analyses require the solution to be obtained in a step-by-step manner. Therefore the analyst subdivides the process of loading into discrete increments, allowing the prediction of the progressive stiffening (or softening) of the structure as the geometry evolves. Geometrically non-linear analysis can be divided into 4 categories:

1. large displacements with small (infinitesimal) rotations and small strains,

2. large displacements with large rotations and small strains,

3. large displacements with small rotations and large (finite) strains, and

4. large displacements with large rotations and large strains.

This thesis considers the most general case of geometric non-linearity, with large displacements and large rotations in combination with large (finite) strains. Thus, methods such as the *co-rotational* approach are not considered here as it is primarily concerned with large displacements with large rotations and small strains (Felippa and Haugen; 2005). Further to those different categories, there are two alternative approaches to tackling such analyses. These are

1. the approach based on a *material* description (also known as the *Lagrangian* description), where the formulation follows the motion of particular particles within the structure, and

2. the approach based on a *spatial* description (also known as the *Eulerian* description), where the formulation tracks the motion of particles passing through a fixed control volume (Rice; 2017).

The *Lagrangian* approach is often used for the analysis of solids whereas the *Eulerian* approach is largely used for the study of fluid motion. The preference for a *Lagrangian* description for solids is due to the extra computational effort required to update the control volume location as the geometry changes when solving for the deformation of solids in an *Eulerian* approach.

Deformation gradient

A structure, positioned within the *Cartesian space* $\{X\}= \{X, Y, Z\}^T$, can be displaced, rotated and/or deformed. The initial location is defined in terms of the orthogonal rectangular coordinates X, Y and Z at time, $t = 0$. Once the object moves to a new position it occupies an alternative *Cartesian space* defined in terms of the coordinates $\{x\}= \{ x, y, z\}^T$. This movement can be described using (3.32).

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \Big[F\Big] \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} \tag{3.32}$$

where $[F]$ is the deformation gradient, expressed as follows

$$\Big[F\Big] = \begin{bmatrix} \frac{\partial x}{\partial X} & \frac{\partial x}{\partial Y} & \frac{\partial x}{\partial Z} \\ \frac{\partial y}{\partial X} & \frac{\partial y}{\partial Y} & \frac{\partial y}{\partial Z} \\ \frac{\partial z}{\partial X} & \frac{\partial z}{\partial Y} & \frac{\partial z}{\partial Z} \end{bmatrix} \tag{3.33}$$

Note that not all displacements acting on an object will cause the object to experience stress or strain; rigid body translation and/or rotation induce no straining.

Consider material particles in an object, where P_2 is situated next to another material particle, P_1 as shown in Figure 3.5. The position of particle P_2 can be described relative to the material particle P_1 as shown in (3.34).

$$\Big\{\Delta X\Big\} = \Big\{X_{P_2}\Big\} - \Big\{X_{P_1}\Big\} \tag{3.34}$$

where $\{\Delta X\} = \{\Delta X, \Delta Y, \Delta Z\}^T$, $\{X_{P_1}\} = \{X_1, Y_1, Z_1\}^T$ and
$\{X_{P_2}\} = \{X_2, Y_2, Z_2\}^T$. Likewise, the relative distance between the material particles after deformation is given by

$$\Big\{\Delta x\Big\} = \Big\{x_{P_2}\Big\} - \Big\{x_{P_1}\Big\} \tag{3.35}$$

The new positions of the material particles are determined using the deformation gradient as shown below.

$$\Big\{\Delta x_{P_1}\Big\} = \Big[F\Big] \Big\{\Delta X_{P_1}\Big\} \tag{3.36}$$

$$\Big\{\Delta x_{P_2}\Big\} = \Big[F\Big] \Big\{\Delta X_{P_2}\Big\} \tag{3.37}$$

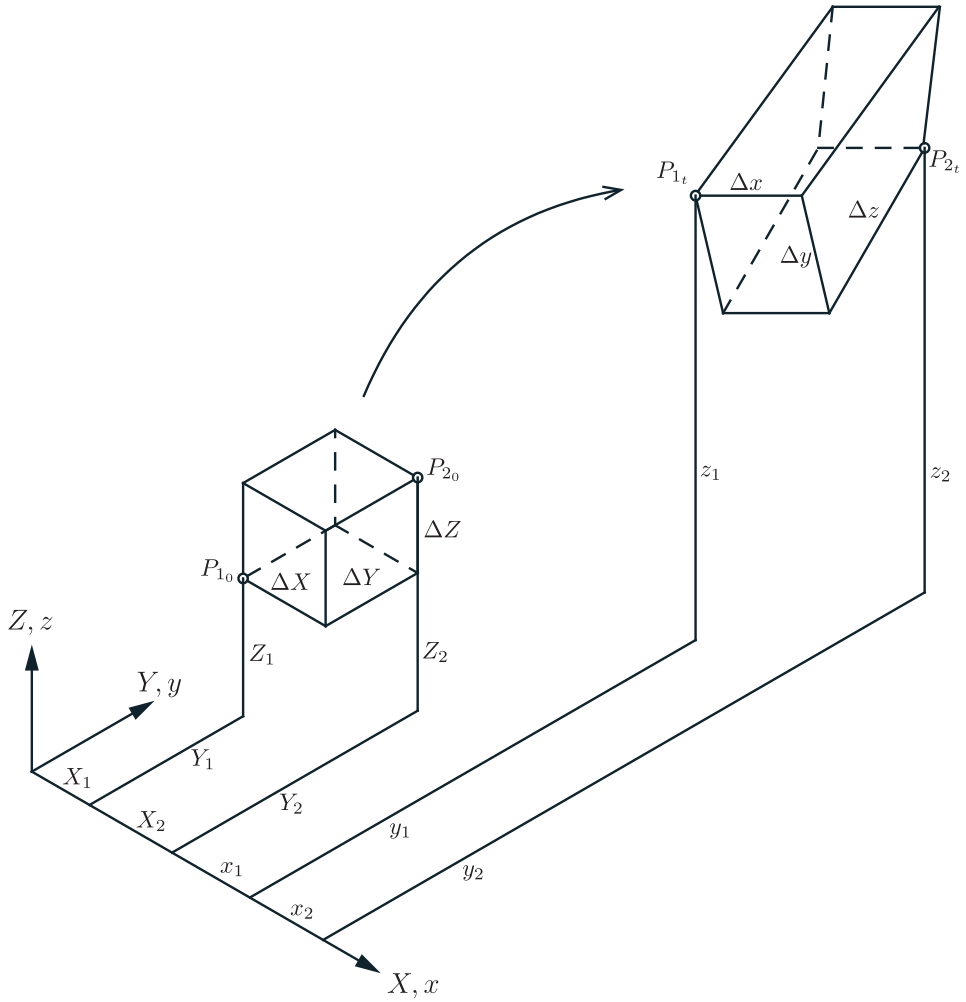


Figure 3.5: Motion of two material particles P_1 and P_2. Note that in this example the original cube is transformed into an irregular hexagon (it is stretched, sheared and flipped over) as a consequence of the displacements.

The changes induced by deformation can be expressed in one of two ways: (i) the change experienced to convert the material particle from its initial position to its deformed position or (ii) the change that reverts the material particle from its deformed position back to its initial position. The concept of strain is employed to express the relative change in distance between two points, or relative rotation, from the initial to the deformed positions.

The *right Cauchy-Green deformation matrix*, $[C]$, is a matrix (or second-order tensor) defined using the deformation gradient, as shown in (3.38).

$$\Big[C\Big] = \Big[F\Big]^T \Big[F\Big] \tag{3.38}$$

The *right Cauchy-Green deformation matrix* is used to define the change in distance caused by the deformation as shown in (3.39).

$$\left\{\Delta x\right\}^T \left\{\Delta x\right\} = \left\{\Delta X\right\}^T \left[C\right] \left\{\Delta X\right\} \tag{3.39}$$

To obtain the reverse effect, that is, to determine the initial distance using the deformed distance and the deformation gradient, the *left Cauchy-Green matrix*, $[b]$ is used, as shown below.

$$\left\{\Delta X\right\}^T \left\{\Delta X\right\} = \left\{\Delta x\right\}^T \left[b\right]^{-1} \left\{\Delta x\right\} \tag{3.40}$$

$$\left[b\right] = \left[F\right] \left[F\right]^T \tag{3.41}$$

Strain measures can be established using each of these deformation matrices as shown in (3.42) and (3.43) respectively, where $[\varepsilon]$ is known as the *Lagrangian-Green strain matrix* and $[e]$ is known as the *Eulerian-Almansi strain matrix*. $[I]$ is the *identity matrix*.

$$\left[\varepsilon\right] = \frac{1}{2}\left[\left[C\right] - \left[I\right]\right] \tag{3.42}$$

$$\left[e\right] = \frac{1}{2}\left[\left[I\right] - \left[b\right]^{-1}\right] \tag{3.43}$$

As the undeformed and deformed configurations can be very different in large deformation analysis. These strain measures creates a distinct frame to reference the material to, which is crucial to ensure work-conjugate stress-strain pairs in the subsections explained later.

Polar decomposition

As mentioned earlier, there are displacements that will cause an object to experience stress or strain and some that doesn't. It is essential to isolate these components. This can be done by further splitting the deformation gradient, $[F]$, into two components: a *stretching* component, $[U]$ and a *rotation* component, $[R]$ as given by (3.44) and illustrated by Figure 3.6.

$$\Big[F\Big] = \Big[R\Big]\Big[U\Big] \tag{3.44}$$

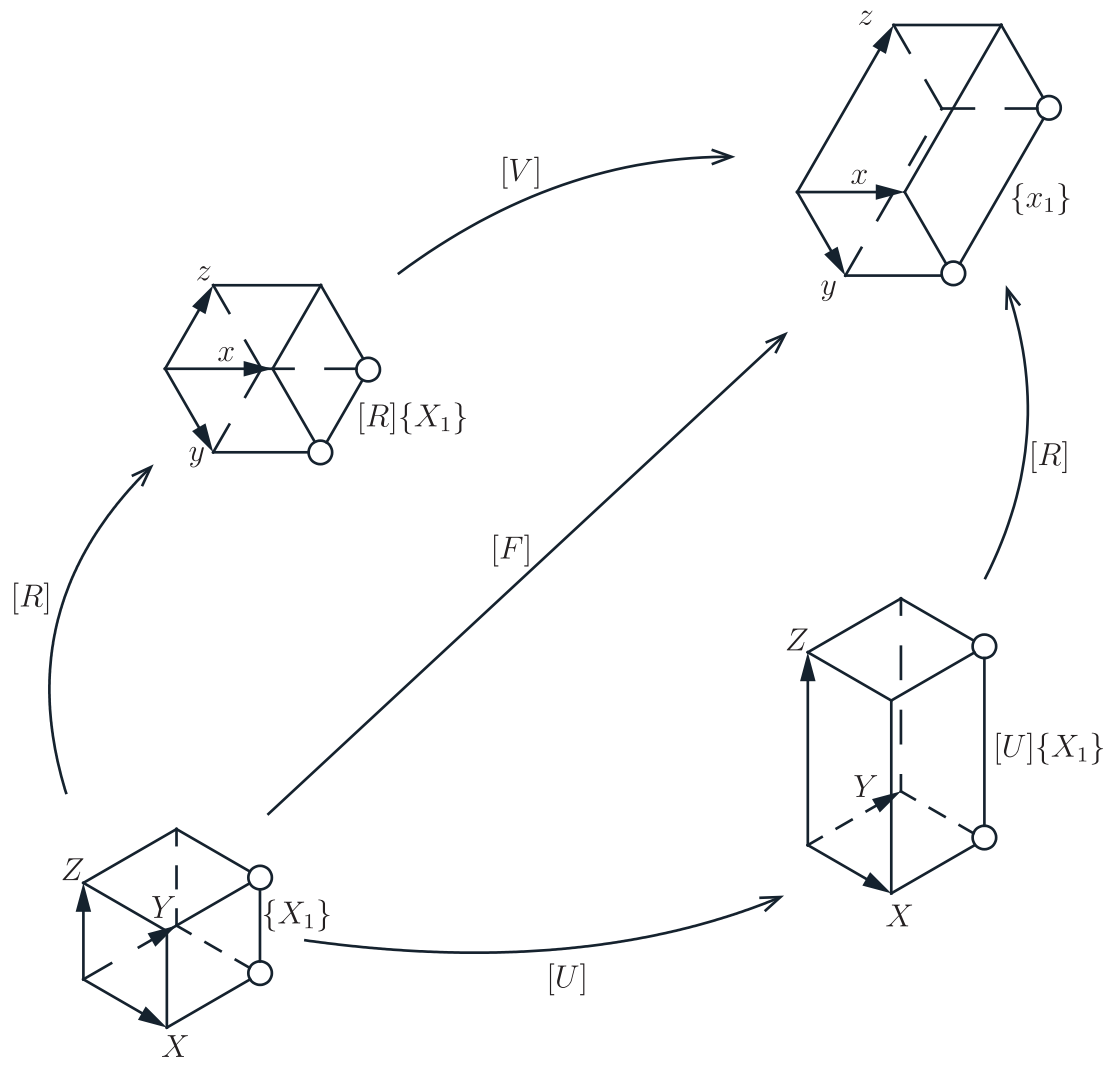


Figure 3.6: Polar decomposition

The *right symmetric stretch matrix* $[U]$ can be determined by substituting (3.44) into (3.38) to obtain (3.45).

$$\Big[C\Big] = \Big[U\Big]^T \Big[R\Big]^T \Big[R\Big] \Big[U\Big] \tag{3.45}$$

Note that the rotation matrix, $[R]$, is orthogonal. Thus $[R]^T[R]$ equals the identity matrix, $[I]$, leading to

$$\Big[C\Big] = \Big[U\Big]^T \Big[U\Big] \tag{3.46}$$

It is also possible to consider the deformation gradient $[F]$ as arising by first rotating the initial configuration by $[R]$, and then stretching $[V]$ the rotated configuration as implied below.

$$\Big[F\Big] = \Big[V\Big] \Big[R\Big] \tag{3.47}$$

By introducing (3.44) into (3.47) and re-arranging the equation, the *left symmetric stretch matrix*, $[V]$, is defined as

$$\Big[V\Big] = \Big[R\Big] \Big[U\Big] \Big[R\Big]^T \tag{3.48}$$

Total Lagrangian and Updated Lagrangian

In large deformation Lagrangian analyses, there are two ways to account for the deformed geometry: the *Total Lagrangian* or the *Updated Lagrangian* approach. In the former approach, the equilibrium is solved at the reference configuration (that is, the stress and strain states, and the stiffness within the structure are all *pulled back* to the reference configuration). In the *Updated Lagrangian* approach, equilibrium is accounted for in the current configuration. To operate with these different descriptions, new stress and strain measures are required.

In a *Total Lagrangian* approach, the *second Piola-Kirchhoff* stresses, $[S]$[3] stresses

[3] *Second Piola-Kirchhoff* stresses are used because of its work-conjugate with the *Green-Lagrange* strain, which is a property that *first Piola-Kirchhoff* stresses (the stresses that relate the current stresses to the reference configuration) lacks.

are used and are updated as follows

$$ {}^{t+\Delta t}_{0}\Big[S\Big] = \overbrace{{}^{t}_{0}\Big[S\Big]}^{known} + \overbrace{{}^{\Delta t}\Big[\Delta S\Big]}^{increment} \tag{3.49} $$

where the zero subscript proceeding $\Big[S\Big]$ indicates that we are referring to the reference configuration and the preceeding superscript identifies the *time-step* under consideration. The strain used in the *Total Lagrangian* approach is the *Green-Lagrange* strain, which can be expressed as follows (Bathe; 2014)[4]

$$ \begin{Bmatrix} \varepsilon_{XX} \\ \varepsilon_{YY} \\ \varepsilon_{ZZ} \\ 2\varepsilon_{XY} \\ 2\varepsilon_{YZ} \\ 2\varepsilon_{XZ} \end{Bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial X} + \frac{1}{2}\Big(\big(\frac{\partial u}{\partial X}\big)^2 + \big(\frac{\partial v}{\partial X}\big)^2 + \big(\frac{\partial w}{\partial X}\big)^2\Big) \\ \frac{\partial v}{\partial Y} + \frac{1}{2}\Big(\big(\frac{\partial u}{\partial Y}\big)^2 + \big(\frac{\partial v}{\partial Y}\big)^2 + \big(\frac{\partial w}{\partial Y}\big)^2\Big) \\ \frac{\partial w}{\partial Z} + \frac{1}{2}\Big(\big(\frac{\partial u}{\partial Z}\big)^2 + \big(\frac{\partial v}{\partial Z}\big)^2 + \big(\frac{\partial w}{\partial Z}\big)^2\Big) \\ \frac{1}{2}\Big(\frac{\partial u}{\partial Y} + \frac{\partial v}{\partial X} + \frac{\partial u}{\partial X}\frac{\partial u}{\partial Y} + \frac{\partial v}{\partial X}\frac{\partial v}{\partial Y} + \frac{\partial w}{\partial X}\frac{\partial w}{\partial Y}\Big) \\ \frac{1}{2}\Big(\frac{\partial w}{\partial Y} + \frac{\partial v}{\partial Z} + \frac{\partial u}{\partial Z}\frac{\partial u}{\partial Y} + \frac{\partial v}{\partial Z}\frac{\partial v}{\partial Y} + \frac{\partial w}{\partial Z}\frac{\partial w}{\partial Y}\Big) \\ \frac{1}{2}\Big(\frac{\partial u}{\partial Z} + \frac{\partial w}{\partial X} + \frac{\partial u}{\partial X}\frac{\partial u}{\partial Z} + \frac{\partial v}{\partial X}\frac{\partial v}{\partial Z} + \frac{\partial w}{\partial X}\frac{\partial w}{\partial Z}\Big) \end{bmatrix} \tag{3.50} $$

This is updated as follows

$$ {}^{t+\Delta t}_{0}\Big[\varepsilon\Big] = \overbrace{{}^{t}_{0}\Big[\varepsilon\Big]}^{known} + \overbrace{{}^{\Delta t}\Big[\Delta\varepsilon\Big]}^{increment} \tag{3.51} $$

The increment in stress can be determined as

$$ {}^{\Delta t}\Big[\Delta S\Big] = \Big[D\Big]\,{}^{\Delta t}\Big[\Delta\varepsilon\Big] \tag{3.52} $$

The components of the strain increments[5] are given by

[4]Note that the shear strain terms correspond to the engineering shear strains rather than the tensorial shear strains.

[5]Note that $\{\epsilon\} = \{\varepsilon_{XX}, \varepsilon_{YY}, \varepsilon_{ZZ}, 2\varepsilon_{XY}, 2\varepsilon_{YZ}, 2\varepsilon_{ZZ}\}^T$, which may be re-written as $[\varepsilon] = \begin{bmatrix} \varepsilon_{XX} & \varepsilon_{XY} & \varepsilon_{XZ} \\ \varepsilon_{YX} & \varepsilon_{YY} & \varepsilon_{YZ} \\ \varepsilon_{ZX} & \varepsilon_{ZY} & \varepsilon_{ZZ} \end{bmatrix}$.

$$ {}^{\Delta t}\Big\{\Delta\varepsilon\Big\} = {}^{\Delta t}\Big\{\Delta\varepsilon_{lin}\Big\} + {}^{\Delta t}\Big\{\Delta\varepsilon_{ini}\Big\} + {}^{\Delta t}\Big\{\Delta\varepsilon_{nli}\Big\} \tag{3.53} $$

where ${}^{\Delta t}\{\varepsilon_{lin}\}$ are the linear components of the strain increments, ${}^{\Delta t}\{\varepsilon_{ini}\}$ account for the initial displacement effect[6] and ${}^{\Delta t}\{\varepsilon_{nli}\}$ are the non-linear components of the strain increments as given below

$$ {}^{\Delta t}\Big\{\Delta\varepsilon_{lin}\Big\} = \begin{Bmatrix} \frac{\partial\Delta u}{\partial X} \\ \frac{\partial\Delta v}{\partial X} \\ \frac{\partial\Delta w}{\partial X} \\ \frac{\partial\Delta u}{\partial Y}\frac{\partial\Delta v}{\partial X} \\ \frac{\partial\Delta v}{\partial Z}\frac{\partial\Delta w}{\partial Y} \\ \frac{\partial\Delta u}{\partial Z}\frac{\partial\Delta w}{\partial X} \end{Bmatrix} \tag{3.54} $$

$$ {}^{\Delta t}\Big\{\Delta\varepsilon_{ini}\Big\} = \frac{1}{2}\begin{Bmatrix} \frac{\partial\,{}^{t}u}{\partial X}\frac{\partial\Delta u}{\partial X} + \frac{\partial\Delta u}{\partial X}\frac{\partial\,{}^{t}u}{\partial X} + \frac{\partial\,{}^{t}v}{\partial X}\frac{\partial\Delta v}{\partial X} + \frac{\partial\Delta v}{\partial X}\frac{\partial\,{}^{t}v}{\partial X} + \frac{\partial\,{}^{t}w}{\partial X}\frac{\partial\Delta w}{\partial X} + \frac{\partial\Delta w}{\partial X}\frac{\partial\,{}^{t}w}{\partial X} \\ \frac{\partial\,{}^{t}u}{\partial Y}\frac{\partial\Delta u}{\partial Y} + \frac{\partial\Delta u}{\partial Y}\frac{\partial\,{}^{t}u}{\partial Y} + \frac{\partial\,{}^{t}v}{\partial Y}\frac{\partial\Delta v}{\partial Y} + \frac{\partial\Delta v}{\partial Y}\frac{\partial\,{}^{t}v}{\partial Y} + \frac{\partial\,{}^{t}w}{\partial Y}\frac{\partial\Delta w}{\partial Y} + \frac{\partial\Delta w}{\partial Y}\frac{\partial\,{}^{t}w}{\partial Y} \\ \frac{\partial\,{}^{t}u}{\partial Z}\frac{\partial\Delta u}{\partial Z} + \frac{\partial\Delta u}{\partial Z}\frac{\partial\,{}^{t}u}{\partial Z} + \frac{\partial\,{}^{t}v}{\partial Z}\frac{\partial\Delta v}{\partial Z} + \frac{\partial\Delta v}{\partial Z}\frac{\partial\,{}^{t}v}{\partial Z} + \frac{\partial\,{}^{t}w}{\partial Z}\frac{\partial\Delta w}{\partial Z} + \frac{\partial\Delta w}{\partial Z}\frac{\partial\,{}^{t}w}{\partial Z} \\ \frac{\partial\,{}^{t}u}{\partial X}\frac{\partial\Delta u}{\partial Y} + \frac{\partial\Delta u}{\partial Y}\frac{\partial\,{}^{t}u}{\partial X} + \frac{\partial\,{}^{t}v}{\partial X}\frac{\partial\Delta v}{\partial Y} + \frac{\partial\Delta v}{\partial Y}\frac{\partial\,{}^{t}v}{\partial X} + \frac{\partial\,{}^{t}w}{\partial X}\frac{\partial\Delta w}{\partial Y} + \frac{\partial\Delta w}{\partial Y}\frac{\partial\,{}^{t}w}{\partial X} \\ \frac{\partial\,{}^{t}u}{\partial Y}\frac{\partial\Delta u}{\partial Z} + \frac{\partial\Delta u}{\partial Z}\frac{\partial\,{}^{t}u}{\partial Y} + \frac{\partial\,{}^{t}v}{\partial Y}\frac{\partial\Delta v}{\partial Z} + \frac{\partial\Delta v}{\partial Z}\frac{\partial\,{}^{t}v}{\partial Y} + \frac{\partial\,{}^{t}w}{\partial Y}\frac{\partial\Delta w}{\partial Z} + \frac{\partial\Delta w}{\partial Z}\frac{\partial\,{}^{t}w}{\partial Y} \\ \frac{\partial\,{}^{t}u}{\partial X}\frac{\partial\Delta u}{\partial Z} + \frac{\partial\Delta u}{\partial Z}\frac{\partial\,{}^{t}u}{\partial X} + \frac{\partial\,{}^{t}v}{\partial X}\frac{\partial\Delta v}{\partial Z} + \frac{\partial\Delta v}{\partial Z}\frac{\partial\,{}^{t}v}{\partial X} + \frac{\partial\,{}^{t}w}{\partial X}\frac{\partial\Delta w}{\partial Z} + \frac{\partial\Delta w}{\partial Z}\frac{\partial\,{}^{t}w}{\partial X} \end{Bmatrix} \tag{3.55} $$

$$ {}^{\Delta t}\Big\{\Delta\varepsilon_{nli}\Big\} = \frac{1}{2}\begin{Bmatrix} \frac{\partial\Delta u}{\partial X}\frac{\partial\Delta u}{\partial X} + \frac{\partial\Delta v}{\partial X}\frac{\partial\Delta v}{\partial X} + \frac{\partial\Delta w}{\partial X}\frac{\partial\Delta w}{\partial X} \\ \frac{\partial\Delta u}{\partial Y}\frac{\partial\Delta u}{\partial Y} + \frac{\partial\Delta v}{\partial Y}\frac{\partial\Delta v}{\partial Y} + \frac{\partial\Delta w}{\partial Y}\frac{\partial\Delta w}{\partial Y} \\ \frac{\partial\Delta u}{\partial Z}\frac{\partial\Delta u}{\partial Z} + \frac{\partial\Delta v}{\partial Z}\frac{\partial\Delta v}{\partial Z} + \frac{\partial\Delta w}{\partial Z}\frac{\partial\Delta w}{\partial Z} \\ \frac{\partial\Delta u}{\partial X}\frac{\partial\Delta u}{\partial Y} + \frac{\partial\Delta v}{\partial X}\frac{\partial\Delta v}{\partial Y} + \frac{\partial\Delta w}{\partial X}\frac{\partial\Delta w}{\partial Y} \\ \frac{\partial\Delta u}{\partial Y}\frac{\partial\Delta u}{\partial Z} + \frac{\partial\Delta v}{\partial Y}\frac{\partial\Delta v}{\partial Z} + \frac{\partial\Delta w}{\partial Y}\frac{\partial\Delta w}{\partial Z} \\ \frac{\partial\Delta u}{\partial X}\frac{\partial\Delta u}{\partial Z} + \frac{\partial\Delta v}{\partial X}\frac{\partial\Delta v}{\partial Z} + \frac{\partial\Delta w}{\partial X}\frac{\partial\Delta w}{\partial Z} \end{Bmatrix} \tag{3.56} $$

where ${}^{t}u$ is the x-displacement at time t. In a geometrically non-linear analysis, the

[6]Note that the strain in *Total Lagrangian* approach needs to relate to the reference configuration. The strain has to be compensated accordingly to account for the potentially different orientations of the reference configuration and the configuration associated with the current time-step.

stiffness matrix has to take account of the initial displacement effect. Recall that the stiffness of a geometrically linear structure is expressed by (3.30). This is expanded to include the geometrically non-linear components as follows

$$\left[K^{TL}\right] = \left[K_L^{TL}\right] + \left[K_{NL}^{TL}\right] \tag{3.57}$$

where $[K^{TL}]$ is the stiffness matrix associated with the *Total Lagrangian* approach, with $[K_L^{TL}]$ being the linear component and $[K_{NL}^{TL}]$ being the non-linear component. The linear components of the *Total Lagrangian* stiffness matrix are similar in all respects to those of the geometrically linear case except that the $[B]$ in (3.30) no longer applies, as the deformed configuration has to be taken into consideration. This results in the need to construct the $[B^{TL}]$ from linear and non-linear components as follows

$$\left[B^{TL}\right] = \left[B_L^{TL}\right] + \left[B_{NL}^{TL}\right] \tag{3.58}$$

Note that $[B_L^{TL}]$ is as given in (3.30). The non-linear strain components could be written as

$$^{t+\Delta t}\left\{\varepsilon_{nli}\right\} = \frac{1}{2}\begin{Bmatrix} \frac{\partial^{t+\Delta t}u}{\partial X}\frac{\partial^{t+\Delta t}u}{\partial X} + \frac{\partial^{t+\Delta t}v}{\partial X}\frac{\partial^{t+\Delta t}v}{\partial X} + \frac{\partial^{t+\Delta t}w}{\partial X}\frac{\partial^{t+\Delta t}w}{\partial X} \\ \frac{\partial^{t+\Delta t}u}{\partial Y}\frac{\partial^{t+\Delta t}u}{\partial Y} + \frac{\partial^{t+\Delta t}v}{\partial Y}\frac{\partial^{t+\Delta t}v}{\partial Y} + \frac{\partial^{t+\Delta t}w}{\partial Y}\frac{\partial^{t+\Delta t}w}{\partial Y} \\ \frac{\partial^{t+\Delta t}u}{\partial Z}\frac{\partial^{t+\Delta t}u}{\partial Z} + \frac{\partial^{t+\Delta t}v}{\partial Z}\frac{\partial^{t+\Delta t}v}{\partial Z} + \frac{\partial^{t+\Delta t}w}{\partial Z}\frac{\partial^{t+\Delta t}w}{\partial Z} \\ \frac{\partial^{t+\Delta t}u}{\partial X}\frac{\partial^{t+\Delta t}u}{\partial Y} + \frac{\partial^{t+\Delta t}v}{\partial X}\frac{\partial^{t+\Delta t}v}{\partial Y} + \frac{\partial^{t+\Delta t}w}{\partial X}\frac{\partial^{t+\Delta t}w}{\partial Y} \\ \frac{\partial^{t+\Delta t}u}{\partial Y}\frac{\partial^{t+\Delta t}u}{\partial Z} + \frac{\partial^{t+\Delta t}v}{\partial Y}\frac{\partial^{t+\Delta t}v}{\partial Z} + \frac{\partial^{t+\Delta t}w}{\partial Y}\frac{\partial^{t+\Delta t}w}{\partial Z} \\ \frac{\partial^{t+\Delta t}u}{\partial X}\frac{\partial^{t+\Delta t}u}{\partial Z} + \frac{\partial^{t+\Delta t}v}{\partial X}\frac{\partial^{t+\Delta t}v}{\partial Z} + \frac{\partial^{t+\Delta t}w}{\partial X}\frac{\partial^{t+\Delta t}w}{\partial Z} \end{Bmatrix} \tag{3.59}$$

which could be re-expressed as

$$
{}^{t+\Delta t}\left\{\varepsilon_{nli}\right\} = \frac{1}{2}\left[A\right]\left[G\right]\begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ \vdots \\ u_8 \\ v_8 \\ w_8 \end{Bmatrix} \tag{3.60}
$$

where

$$
\left[A\right] = \begin{bmatrix} \left\{\theta_X\right\}^T & \left\{0\right\}^T & \left\{0\right\}^T \\ \left\{0\right\}^T & \left\{\theta_Y\right\}^T & \left\{0\right\}^T \\ \left\{0\right\}^T & \left\{0\right\}^T & \left\{\theta_Z\right\}^T \\ \left\{\theta_Y\right\}^T & \left\{\theta_X\right\}^T & \left\{0\right\}^T \\ \left\{0\right\}^T & \left\{\theta_Z\right\}^T & \left\{\theta_Y\right\}^T \\ \left\{\theta_Z\right\}^T & \left\{0\right\}^T & \left\{\theta_X\right\}^T \end{bmatrix} \tag{3.61}
$$

$$
\left\{\theta_X\right\}^T = \left\{\frac{\partial^{t+\Delta t}u}{\partial X} \quad \frac{\partial^{t+\Delta t}v}{\partial X} \quad \frac{\partial^{t+\Delta t}w}{\partial X}\right\}
$$

$$
\left\{\theta_Y\right\}^T = \left\{\frac{\partial^{t+\Delta t}u}{\partial Y} \quad \frac{\partial^{t+\Delta t}v}{\partial Y} \quad \frac{\partial^{t+\Delta t}w}{\partial Y}\right\}
$$

$$
\left\{\theta_Z\right\}^T = \left\{\frac{\partial^{t+\Delta t}u}{\partial Z} \quad \frac{\partial^{t+\Delta t}v}{\partial Z} \quad \frac{\partial^{t+\Delta t}w}{\partial Z}\right\}
$$

$$
\left\{0\right\}^T = \left\{0 \quad 0 \quad 0\right\}
$$

and

$$\Big[G\Big] = \begin{bmatrix} \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & 0 & 0 & \dots & \frac{\partial N_8}{\partial X} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & 0 & \dots & 0 & \frac{\partial N_8}{\partial X} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & \dots & 0 & 0 & \frac{\partial N_8}{\partial X} \\ \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & 0 & 0 & \dots & \frac{\partial N_8}{\partial Y} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & 0 & \dots & 0 & \frac{\partial N_8}{\partial Y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & \dots & 0 & 0 & \frac{\partial N_8}{\partial Y} \\ \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & 0 & 0 & \dots & \frac{\partial N_8}{\partial Z} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & 0 & \dots & 0 & \frac{\partial N_8}{\partial Z} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & \dots & 0 & 0 & \frac{\partial N_8}{\partial Z} \end{bmatrix} \tag{3.62}$$

The non-linear strain variation could then be written as[7]

$$^{t+\Delta t}\Big\{\Delta\varepsilon_{nli}\Big\} = \Big[A\Big] \begin{bmatrix} \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & 0 & 0 & \dots & \frac{\partial N_8}{\partial X} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & 0 & \dots & 0 & \frac{\partial N_8}{\partial X} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & \dots & 0 & 0 & \frac{\partial N_8}{\partial X} \\ \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & 0 & 0 & \dots & \frac{\partial N_8}{\partial Y} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & 0 & \dots & 0 & \frac{\partial N_8}{\partial Y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & \dots & 0 & 0 & \frac{\partial N_8}{\partial Y} \\ \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & 0 & 0 & \dots & \frac{\partial N_8}{\partial Z} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & 0 & \dots & 0 & \frac{\partial N_8}{\partial Z} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & \dots & 0 & 0 & \frac{\partial N_8}{\partial Z} \end{bmatrix} \begin{Bmatrix} \Delta u_1 \\ \Delta v_1 \\ \Delta w_1 \\ \Delta u_2 \\ \Delta v_2 \\ \Delta w_2 \\ \vdots \\ \Delta u_8 \\ \Delta v_8 \\ \Delta w_8 \end{Bmatrix} \tag{3.63}$$

Knowing $\{\Delta\varepsilon\} = [B]\{\Delta d\}$, the non-linear $[B_{NL}^{TL}]$ can be expressed as follows (Jensen et al.; 2001)

[7] This is because $^{t+\Delta t}\{\Delta\varepsilon_{nli}\} = \frac{1}{2}[\Delta A][G]\{d\} + \frac{1}{2}[A][G]\{\Delta d\}$. Since $\frac{1}{2}[\Delta A][G]\{d\} = \frac{1}{2}[A][G]\{\Delta d\}$, $^{t+\Delta t}\{\Delta\varepsilon_{nli}\} = [A][G]\{\Delta d\}$

$$\left[B_{NL}^{TL}\right] = \left[A\right] \begin{bmatrix} \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & 0 & 0 & \dots & \frac{\partial N_8}{\partial X} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & 0 & \dots & 0 & \frac{\partial N_8}{\partial X} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial X} & 0 & 0 & \frac{\partial N_2}{\partial X} & \dots & 0 & 0 & \frac{\partial N_8}{\partial X} \\ \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & 0 & 0 & \dots & \frac{\partial N_8}{\partial Y} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & 0 & \dots & 0 & \frac{\partial N_8}{\partial Y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial Y} & 0 & 0 & \frac{\partial N_2}{\partial Y} & \dots & 0 & 0 & \frac{\partial N_8}{\partial Y} \\ \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & 0 & 0 & \dots & \frac{\partial N_8}{\partial Z} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & 0 & \dots & 0 & \frac{\partial N_8}{\partial Z} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial Z} & 0 & 0 & \frac{\partial N_2}{\partial Z} & \dots & 0 & 0 & \frac{\partial N_8}{\partial Z} \end{bmatrix} \tag{3.64}$$

The non-linear stiffness matrix components are expressed as follows

$$\left[K_{NL}^{TL}\right] = \begin{bmatrix} \left[K_{NL}^{TL}\right]_{11} & \left[K_{NL}^{TL}\right]_{12} & \cdots & \left[K_{NL}^{TL}\right]_{1n} \\ \left[K_{NL}^{TL}\right]_{21} & \ddots & & \\ \vdots & & & \\ \left[K_{NL}^{TL}\right]_{n1} & \cdots & & \left[K_{NL}^{TL}\right]_{nn} \end{bmatrix} \tag{3.65}$$

where, for example,

$$\begin{aligned} \left[K_{NL}^{TL}\right]_{11} &= \begin{Bmatrix} \frac{\partial N_1}{\partial X} \\ \frac{\partial N_1}{\partial Y} \\ \frac{\partial N_1}{\partial Z} \end{Bmatrix}^T \left[S\right] \begin{Bmatrix} \frac{\partial N_1}{\partial X} \\ \frac{\partial N_1}{\partial Y} \\ \frac{\partial N_1}{\partial Z} \end{Bmatrix} \left[I\right] \\ \left[K_{NL}^{TL}\right]_{12} &= \begin{Bmatrix} \frac{\partial N_1}{\partial X} \\ \frac{\partial N_1}{\partial Y} \\ \frac{\partial N_1}{\partial Z} \end{Bmatrix}^T \left[S\right] \begin{Bmatrix} \frac{\partial N_2}{\partial X} \\ \frac{\partial N_2}{\partial Y} \\ \frac{\partial N_2}{\partial Z} \end{Bmatrix} \left[I\right] \end{aligned} \tag{3.66}$$

Given that the *Updated Lagrangian* approach refers to the deformed configuration, a new $[B^{UL}]$ will be required. Thus the $[B]$ from (3.6) will need to be modified accordingly, as

$$\left[B^{UL}\right] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_2}{\partial x} & 0 & 0 & \dots & \frac{\partial N_8}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & \dots & 0 & \frac{\partial N_8}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & \dots & 0 & 0 & \frac{\partial N_8}{\partial z} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & 0 & \dots & \frac{\partial N_8}{\partial y} & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial z} & \frac{\partial N_2}{\partial y} & \dots & 0 & \frac{\partial N_8}{\partial z} & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial z} & 0 & \frac{\partial N_2}{\partial x} & \dots & \frac{\partial N_8}{\partial z} & 0 & \frac{\partial N_8}{\partial x} \end{bmatrix} \tag{3.67}$$

The *Cauchy* stress, $[\sigma]$ used in *Updated Lagrangian* approach relates the current stress to the current displacement. The *second Piola-Kirchhoff* stress and the *Cauchy* stress can be related as follows

$$\left[S\right] = det\left[F\right]\left[F\right]^{-1}\left[\sigma\right]\left[F\right]^{-T} \tag{3.68}$$

Unlike the *Total Lagrangian* approach, which makes use of the *Green-Lagrange* strains, the *Updated Lagrangian* approach has two possible strain tensors to choose from, namely the *Almansi strain* or the *logarithmic strain*, with the former strain more suitable for large deformations with small strains (Bathe; 2014). The initial displacement effect on the structural stiffness can be incorporated into the *Updated Lagrangian* ($[K^{UL}]$) approach as follows

$$\left[K^{UL}\right] = \left[K_L^{UL}\right] + \left[K_{NL}^{UL}\right] \tag{3.69}$$

where $[K_L^{UL}]$ gives the linear components of the *Updated Lagrangian* stiffness matrix and $[K_{NL}^{UL}]$ gives the non-linear components of the *Updated Lagrangian* stiffness matrix. The $[K_L^{UL}]$ component can be determined by replacing the $[B]$ in (3.30) with (3.67) and the non-linear components are expressed as

$$\left[K_{NL}^{UL}\right] = \begin{bmatrix} \left[K_{NL}^{UL}\right]_{11} & \left[K_{NL}^{UL}\right]_{12} & \dots & \left[K_{NL}^{UL}\right]_{1n} \\ \left[K_{NL}^{UL}\right]_{21} & \ddots & & \\ \vdots & & & \\ \left[K_{NL}^{UL}\right]_{n1} & \dots & & \left[K_{NL}^{UL}\right]_{nn} \end{bmatrix} \tag{3.70}$$

where, for example,

$$\left[K_{NL}^{UL}\right]_{11} = \left\{\begin{array}{c}\frac{\partial N_1}{\partial x}\\ \frac{\partial N_1}{\partial y}\\ \frac{\partial N_1}{\partial z}\end{array}\right\}^T \left[\sigma\right] \left\{\begin{array}{c}\frac{\partial N_1}{\partial x}\\ \frac{\partial N_1}{\partial y}\\ \frac{\partial N_1}{\partial z}\end{array}\right\} \left[I\right]$$
$$\left[K_{NL}^{UL}\right]_{12} = \left\{\begin{array}{c}\frac{\partial N_1}{\partial x}\\ \frac{\partial N_1}{\partial y}\\ \frac{\partial N_1}{\partial z}\end{array}\right\}^T \left[\sigma\right] \left\{\begin{array}{c}\frac{\partial N_2}{\partial x}\\ \frac{\partial N_2}{\partial y}\\ \frac{\partial N_2}{\partial z}\end{array}\right\} \left[I\right] \tag{3.71}$$

This thesis will be working primarily with the *Total Lagrangian* approach, but the *Updated Lagrangian* approach is also used to illustrate other features later in this chapter.

Finding the solutions in a non-linear FE analysis

In non-linear FE analysss, the solution for a given problem can no longer be solved in one step but requires an iterative procedure in order to account for the changing stiffness and forces such that equilibrium can be attained at each stage of the computation. This is achieved by balancing the internal forces (resulting from the deformation) with the external applied forces. The elemental internal forces, $\{f_{int}\}$, are determined using (3.72).

$$\left\{f_{int}\right\} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left[B^{TL}\right]^T \left\{S\right\} \omega_i\omega_j\omega_k det\left[J\right] \tag{3.72}$$

The difference between the external forces applied to the element and the internal forces arising from the internal stresses (also known as *out-of-balance* forces $\{f_{oob}\}$) is given by

$$\left\{f_{oob}\right\} = \left\{f_{ext}\right\} + \left\{f_{react}\right\} - \left\{f_{int}\right\} \tag{3.73}$$

where $\{f_{react}\}$ are the reaction forces which can be determined as follows

$$\left\{f_{react}\right\} = \left[K\right]_{bc}\left\{d\right\}_{bc} - \left\{f_{oob}\right\} \tag{3.74}$$

In (3.74) $[K]_{bc}$ is the stiffness matrix consisting only of components associated with the known boundary conditions and $\{d\}_{bc}$ are the displacements at those boundary locations. If the out of balance force is within a pre-defined tolerance, then the load step will have been completed and the analysis can move on to the next load step. However, if it is not within the tolerance, then the displacements requires further updating $\{\Delta u\}$ through use of the tangent stiffness matrix, $[K]$ (determined earlier) as shown below

$$\left[K\right]\left\{\Delta d\right\} = \left\{f_{oob}\right\} \tag{3.75}$$

Example of a geometrically non-linear finite element algorithm

Listing 3.2[8] shows an example of a geometrically non-linear analysis code using the *Total Lagrangian* approach. The flow diagram is shown in Figure 3.7. The number in the top left corner of a box refers to the line number in Listing 3.2 whereas the number in the top right corner of a box refers to the line number in Listing 3.3.

```
1  [coord,etpl,fext0,bc,D,ngp,lstps,NRitmax,NRtol]=...
2    GNLcantilever_endload;
3  [nels,nen]=size(etpl); nodes=size(coord,1);
4  ndim=3; nDoF=nodes*ndim; neDoF=(nen*ndim)^2;
5  krow=zeros(neDoF*nels,1); kcol=krow; kval=krow;
6  uvw=zeros(nDoF,1); uvwold=uvw; fint=uvw; react=uvw;
7  fd=(1:nDoF); fd(bc(:,1))=[];
8  epsE=zeros(6,ngp,nels); epsEn=epsE; sig=epsE; sigN=epsE;
9  oduX=zeros(3,3,ngp,nels); duX=oduX;
10 for lstp=0:lstps
11   fext=(lstp/lstps)*fext0; oobf=react+fext-fint;
12   oobfnorm=2*NRtol; NRit=0;
13   while ((NRit<NRitmax)&&(oobfnorm>NRtol))
14     NRit=NRit+1; fint=zeros(nDoF,1); dreact=fint; dduvw=fint;
15     if lstp>=1
16       Kt=sparse(krow,kcol,kval,nDoF,nDoF);
17       dduvw(bc(:,1))=(1+sign(1-NRit))*bc(:,2)/lstps;
18       dduvw(fd)=Kt(fd,fd)\(oobf(fd)-Kt(fd,bc(:,1))*dduvw(bc(:,1)));
19       dreact(bc(:,1))=Kt(bc(:,1),:)*dduvw-oobf(bc(:,1));
20     end
21     uvw=uvw+dduvw; react=react+dreact; duvw=uvw-uvwold;
22     for nel=1:nels
23       ed=reshape(ones(ndim,1)*etpl(nel,:)*...
24                  ndim-(ndim-1:-1:0).'*ones(1,nen),1,nen*ndim);
```

[8]Note that this algorithm draws heavily upon the code published by (Coombs et al.; 2010).

```
25        [ke,felem,epsE(:,:,nel),sig(:,:,nel),duX(:,:,:,nel)]=...
26          TLFE(coord(etpl(nel,:),:),D(:,:,:,nel),duvw(ed),ngp,...
27               epsEn(:,:,nel),sigN(:,:,nel),oduX(:,:,:,nel));
28        if lstp==0; krow((nel-1)*neDoF+1:nel*neDoF)=...
29                      reshape(ed.'*ones(1,nen*ndim),neDoF,1);
30                    kcol((nel-1)*neDoF+1:nel*neDoF)=...
31                      reshape(ones(nen*ndim,1)*ed  ,neDoF,1);
32        end
33        kval((nel-1)*neDoF+1:nel*neDoF)=reshape(ke,neDoF,1);
34        fint(ed)=fint(ed)+felem;
35      end
36      oobf=fext+react-fint; oobfnorm=norm(oobf)/norm(fext+react+eps);
37      fprintf('%4i %4i %6.3e\n',lstp,NRit,oobfnorm);
38    end
39    uvwold=uvw; epsEn=epsE; sigN=sig; oduX=oduX+duX;
40  end
```

Listing 3.2: 40 line MATLAB script expressing the 3D *Total Lagrangian* FE method.

```
function[ke,fint,epsE,sig,duX]=...
  TLFE(coord,D,duvw,ngp,epsEn,sigN,oduX)
nen=size(coord,1); epsE=zeros(6,ngp); sig=epsE;
[wp,GpLoc]=GpPos(ngp); ke=zeros(nen*3); Ks=ke;
fint=zeros(nen*3,1); duX=zeros(3,3,ngp);
for gp=1:ngp
  xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zet=GpLoc(gp,3);
  dNr=dershapefunc2D(xsi,eta,zet);
  dXr=dNr*coord; detJ=det(dXr); dNX=dXr\dNr;
  duX(:,:,gp)=dNX*reshape(duvw,3,nen)';
  epsEt=[0.5*(duX(1,1,gp) + duX(1,1,gp) + ...
         oduX(1,1,gp)*duX(1,1,gp) + oduX(1,2,gp)*duX(1,2,gp) + ...
         oduX(1,3,gp)*duX(1,3,gp) + duX(1,1,gp)*oduX(1,1,gp) + ...
         duX(1,2,gp)*oduX(1,2,gp) + duX(1,3,gp)*oduX(1,3,gp) + ...
         duX(1,1,gp)*duX(1,1,gp) + duX(1,2,gp)*duX(1,2,gp) + ...
         duX(1,3,gp)*duX(1,3,gp));
         0.5*(duX(2,2,gp) + duX(2,2,gp) + ...
         oduX(2,1,gp)*duX(2,1,gp) + oduX(2,2,gp)*duX(2,2,gp) + ...
         oduX(2,3,gp)*duX(2,3,gp) + duX(2,1,gp)*oduX(2,1,gp) + ...
         duX(2,2,gp)*oduX(2,2,gp) + duX(2,3,gp)*oduX(2,3,gp) + ...
         duX(2,1,gp)*duX(2,1,gp) + duX(2,2,gp)*duX(2,2,gp) + ...
         duX(2,3,gp)*duX(2,3,gp));
         0.5*(duX(3,3,gp) + duX(3,3,gp) + ...
         oduX(3,1,gp)*duX(3,1,gp) + oduX(3,2,gp)*duX(3,2,gp) + ...
         oduX(3,3,gp)*duX(3,3,gp) + duX(3,1,gp)*oduX(3,1,gp) + ...
         duX(3,2,gp)*oduX(3,2,gp) + duX(3,3,gp)*oduX(3,3,gp) + ...
         duX(3,1,gp)*duX(3,1,gp) + duX(3,2,gp)*duX(3,2,gp) + ...
         duX(3,3,gp)*duX(3,3,gp));
         1.0*(duX(2,1,gp) + duX(1,2,gp) + ...
         oduX(1,1,gp)*duX(2,1,gp) + oduX(1,2,gp)*duX(2,2,gp) + ...
         oduX(1,3,gp)*duX(2,3,gp) + duX(1,1,gp)*oduX(2,1,gp) + ...
         duX(1,2,gp)*oduX(2,2,gp) + duX(1,3,gp)*oduX(2,3,gp) + ...
         duX(1,1,gp)*duX(2,1,gp) + duX(1,2,gp)*duX(2,2,gp) + ...
         duX(1,3,gp)*duX(2,3,gp));
         1.0*(duX(2,3,gp) + duX(3,2,gp) + ...
         oduX(3,1,gp)*duX(2,1,gp) + oduX(3,2,gp)*duX(2,2,gp) + ...
         oduX(3,3,gp)*duX(2,3,gp) + duX(3,1,gp)*oduX(2,1,gp) + ...
         duX(3,2,gp)*oduX(2,2,gp) + duX(3,3,gp)*oduX(2,3,gp) + ...
         duX(3,1,gp)*duX(2,1,gp) + duX(3,2,gp)*duX(2,2,gp) + ...
         duX(3,3,gp)*duX(2,3,gp));
         1.0*(duX(3,1,gp) + duX(1,3,gp) + ...
         oduX(1,1,gp)*duX(3,1,gp) + oduX(1,2,gp)*duX(3,2,gp) + ...
         oduX(1,3,gp)*duX(3,3,gp) + duX(1,1,gp)*oduX(3,1,gp) + ...
         duX(1,2,gp)*oduX(3,2,gp) + duX(1,3,gp)*oduX(3,3,gp) + ...
         duX(1,1,gp)*duX(3,1,gp) + duX(1,2,gp)*duX(3,2,gp) + ...
         duX(1,3,gp)*duX(3,3,gp))];
  epsE(:,gp)=epsEt+epsEn(:,gp);
  sig(:,gp)=(D(:,:,gp)*epsEt)+sigN(:,gp);
  s=[sig(1,gp) sig(4,gp) sig(6,gp);
     sig(4,gp) sig(2,gp) sig(5,gp);
     sig(6,gp) sig(5,gp) sig(3,gp)];
  BL=formB(dNX,nen); BNL=zeros(6,nen*3);
  tduX=oduX(:,:,gp)+duX(:,:,gp);
  A=[ tduX(1,:) zeros(1,3) zeros(1,3);
     zeros(1,3)  tduX(2,:) zeros(1,3);
     zeros(1,3) zeros(1,3)  tduX(3,:);
      tduX(2,:)  tduX(1,:) zeros(1,3);
     zeros(1,3)  tduX(3,:)  tduX(2,:);
      tduX(3,:) zeros(1,3)  tduX(1,:)];
  for n=1:nen
    for m=1:nen
      Ks(n*3-2:n*3,m*3-2:m*3)=dNX(:,n)'*s*dNX(:,m)*eye(3);
    end
    BNL(:,(n-1)*3+1:(n-1)*3+3)=...
```

```
65        A*[dNX(1,n)*eye(3); dNX(2,n)*eye(3); dNX(3,n)*eye(3)];
66    end
67    Bt=BL+BNL;
68    ke=ke+(Bt'*D(:,:,gp)*Bt+Ks)*detJ*wp(gp);
69    fint=fint+Bt'*sig(:,gp)*detJ*wp(gp);
70 end
```

Listing 3.3: MATLAB function file required for the *Total Lagrangian* analysis, with the main code shown in Listing 3.2.

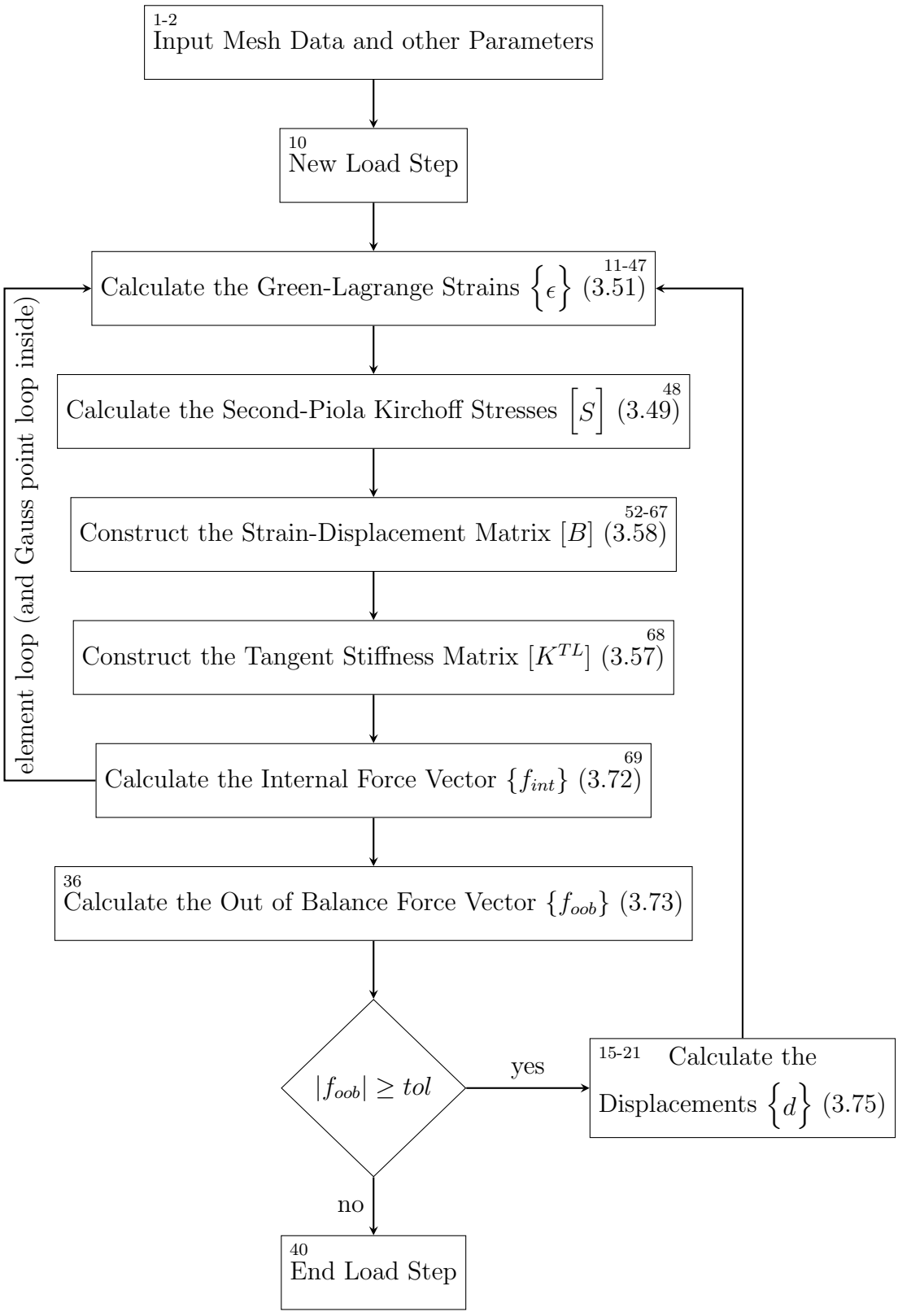


Figure 3.7: Flow diagram depicting the sequential operation within a single load step for the *Total Lagrangian* FE analysis.

Note that the *Input Mesh Data and other Parameters* in the *Total Lagrangian* ap-

proach includes the definition of the coordinates of the nodes (`coord`), the topologies (node listings) of the elements (`etpl`), the external forces applied (`fext0`), the boundary conditions (`bc`), the constitutive matrix (`D`), the number of *Gauss points* (`ngp`), the number of load steps (`lstps`), the maximum number of *Newton-Raphson* iterations allowed per load step (`NRitmax`) and the normalised *Newton-Raphson* tolerance (`NRtol`).

The input format for the data is given in the Table below.

Variable	Matrix Sizing	Format
`coord`	$nen \times 3$	$\begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}$ (in m)
`etpl`	$nels \times nen$	$\begin{bmatrix} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 & n_7 & n_8 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$
`fext0`	$(nen \times dof) \times 1$	$\begin{Bmatrix} fext0_1 \\ fext0_2 \\ \vdots \end{Bmatrix}$ (in N)
`bc`	$nbc \times 2$	$\begin{bmatrix} node\_number & node\_displacement \\ \vdots & \vdots \end{bmatrix}$
`D`	$6 \times 6 \times ngp \times nels$	4-D matrix (in Pa)
`ngp`	Scalar	ngp
`lstps`	Scalar	$lstps$
`NRitmax`	Scalar	$NRitmax$
`NRtol`	Scalar	$NRtol$

Table 3.6: Input format for the 8-noded hexahedral geometrically non-linear analysis, with example values for `ngp`, `lstps`, `NRitmax` and `NRtol`.

where nen is the number of nodes defining each element, dof gives the number of degrees of freedom per node, $nels$ is the total number of elements and nbc is the number of degrees of freedom with displacement constraints. Note that the stresses and strains always relate to the reference configuration in the *Total Lagrangian* approach. This allows the constitutive matrix (see (3.12)) to be pre-computed (for every *Gauss point* in every element) at the input stage as it does not need to be updated during the analysis.

The *Gauss point* and *element* loops within a load step are shown in Figure 3.7. The algorithm would first enter an element, loop through all of its integration points,

Variable	Matrix Sizing	Format
`uvw`	$(nen \times dof) \times 1$	$\left\{ \begin{matrix} u_1 \\ v_1 \\ \vdots \end{matrix} \right\}$ (in metres)

Table 3.7: Output from the non-linear 8-noded hexahedral analysis.

and then move on to the next element (in order to loop through all of that element's integration points once again).

Geometrically non-linear FE numerical example

A very slender cantilever subjected to a transverse end load (Sze et al.; 2004) was used to assess the accuracy of the Total Lagrangian geometrically non-linear FE code.

Figure 3.8: Cantilever subjected to end point load.

Length, L	10 m
Width, b	1 m
Depth, d	0.1 m
Young's modulus, E	1.2 MPa
Poisson's ratio, ν	0
Force applied, F	4 kN

Table 3.8: Input parameters for the non-linear 8-noded hexahedral cantilever.

This benchmark problem was setup as illustrated in Figure 3.8 with the parameters given in Table 3.8. The problem was simulated using 500 8-noded hexahedral elements employing both Listing 3.1 and Listing 3.2. The geometrically nonlinear results are shown to be in good agreement with the reference solution (within 1%, see Figure 3.9). It is evident that a linear analysis fails to account for the progressive stiffening up of the beam under deformation, resulting in a gross under-prediction of the u displacement and over-prediction of the w displacement.

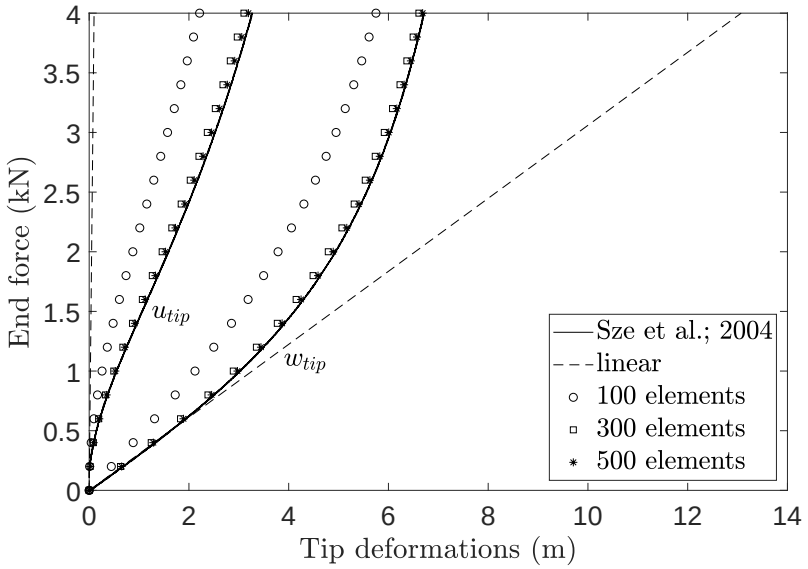


Figure 3.9: Load-deflection curve for the cantilever subjected to an end point load, comparing the linear and geometrically non-linear FE analyses (with different numbers of hexahedral elements) against the analytical solution (Sze et al.; 2004).

Listing 3.2 was further used to determine how the number of elements used influences the accuracy of the solutions. It is seen (in Figure 3.9) that with an increase in the number of elements, the results approach the analytical solution. The input file for the non-linear FE code (Listing 3.2) is shown in Listing II.2. Note that the comparison here was undertaken using 8-noded hexahedral elements which assume the displacement varies linearly along each edge of the element. These elements do not fully capture the deformation unless a significant number are used (in this case, at least 300). It is not believed that locking behaviour is an issue here; it is rather that the large flexure can not be reproduced with a modest mesh of linear elements. A lesser number of elements would be needed to capture this behaviour if 20-noded hexahedral elements (which assume a quadratic variation along each edge of the element) were used.

3.2.2 Material non-linearity

Given sufficient straining, a material will inevitably depart from a linear elastic behaviour such that it experiences a non-linear relationship between the stresses and strains. That is, the material stiffness will change as it undergoes increasing strain.

Material non-linearity comes in many different forms (including for example hypo and hyperelasticity, elasto-plasticity and visco-elasto-plasticity). In this thesis, because of the type of material primarily under examination (the myocardium) and the strain levels experienced, only a *hyperelastic* material model will be considered.

Hyperelasticity

For hyperelastic models, the stresses depend solely on the current state of deformation that the material is experiencing. The *Neo-Hookean* model is widely used for rubber-like materials. In its normal form, it describes a material which stiffens up as the deformation increases. The $[D]$ matrix for this model can be written as (see (7.40) of (Bonet and Wood; 1997))

$$\Big[D\Big] = \begin{bmatrix} \lambda'+2\mu' & \lambda' & \lambda' & 0 & 0 & 0 \\ \lambda' & \lambda'+2\mu' & \lambda' & 0 & 0 & 0 \\ \lambda' & \lambda' & \lambda'+2\mu' & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu' & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu' & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu' \end{bmatrix} \tag{3.76}$$

where $\lambda' = \frac{\lambda}{J}$ and $\mu' = \frac{\mu - \lambda ln(J)}{J}$. Here $J = det[F]$, λ is Lamé's first parameter and μ is the shear modulus. The elastic parameters λ and μ can be expressed in terms of Young's modulus and Poisson's ratio as $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$. The *Cauchy* stresses for such a model are given by (see (5.29) in (Bonet and Wood; 1997))

$$\Big[\sigma\Big] = \frac{\mu}{J}\Big[\Big[b\Big] - \Big[I\Big]\Big] + \frac{\lambda}{J} ln\Big(J\Big)\Big[I\Big] \tag{3.77}$$

Note that to make use of *Cauchy* stresses shown in (3.77), the *Updated Lagrangian* approach has be adopted for the remainder of this chapter. This is consistent with the work of Bonet and Wood (Bonet and Wood; 1997).

3.2.3 Example of a material non-linear finite element algorithm

Listing 3.4 gives an example of a MATLAB script using the *Updated Lagrangian* approach, with the function scripts shown in Listing 3.5.

```
[coord,etpl,fext0,bc,lam,miu,ngp,lstps,NRitmax,NRtol]=CookMembrane;
[nels,nen]=size(etpl); nodes=size(coord,1);
ndim=3; nDoF=nodes*ndim; neDoF=(nen*ndim)^2;
krow=zeros(neDoF*nels,1); kcol=krow; kval=krow;
uvw=zeros(nDoF,1); uvwold=uvw; fint=uvw; react=uvw;
fd=(1:nDoF); fd(bc(:,1))=[];
epsE=zeros(6,ngp,nels); sig=epsE;
Fold=zeros(3,3,ngp,nels); F=Fold;
for n=1:nels; for m=1:ngp; Fold(:,:,m,n)=eye(3); end; end
for lstp=0:lstps
  fext=(lstp/lstps)*fext0; oobf=react+fext-fint;
  oobfnorm=2*NRtol; NRit=0;
  while ((NRit<NRitmax)&&(oobfnorm>NRtol))
    NRit=NRit+1; fint=zeros(nDoF,1); dreact=fint; dduvw=fint;
    if lstp>=1
      Kt=sparse(krow,kcol,kval,nDoF,nDoF);
      dduvw(bc(:,1))=(1+sign(1-NRit))*bc(:,2)/lstps;
      dduvw(fd)=Kt(fd,fd)\(oobf(fd)-Kt(fd,bc(:,1))*dduvw(bc(:,1)));
      dreact(bc(:,1))=Kt(bc(:,1),:)*dduvw-oobf(bc(:,1));
    end
    uvw=uvw+dduvw; react=react+dreact; duvw=uvw-uvwold;
    for nel=1:nels
      ed=reshape(ones(ndim,1)*etpl(nel,:)*ndim-(ndim-1:-1:0)'*...
                                          ones(1,nen),1,nen*ndim);
      [ke,felem,epsE(:,:,nel),sig(:,:,nel),F(:,:,:,nel)]=...
        ULFE(coord(etpl(nel,:),:),lam,miu,uvw(ed),duvw(ed),...
        ngp,Fold(:,:,:,nel));
      if lstp==0; krow((nel-1)*neDoF+1:nel*neDoF)=...
                    reshape(ed.'*ones(1,nen*ndim),neDoF,1);
                  kcol((nel-1)*neDoF+1:nel*neDoF)=...
                    reshape(ones(nen*ndim,1)*ed   ,neDoF,1);
      end
      kval((nel-1)*neDoF+1:nel*neDoF)=reshape(ke,neDoF,1);
      fint(ed)=fint(ed)+felem;
    end
    oobf=fext+react-fint; oobfnorm=norm(oobf)/norm(fext+react+eps);
    fprintf('%4i %4i %6.3e\n',lstp,NRit,oobfnorm);
  end
  uvwold=uvw; Fold=F;
end
```

Listing 3.4: 40 line MATLAB script expressing the 3D *Updated Lagrangian* FE method.

```
function[ke,fint,epsE,sig,F]=...
  ULFE(nodeData,lam,miu,uvw,duvw,ngp,Fold)
nen=size(nodeData,1); coord=nodeData(:,1:3); epsE=zeros(6,ngp);
[wp,GpLoc]=GpPos(ngp); ke=zeros(nen*3); Ks=ke;
fint=zeros(nen*3,1); F=zeros(3,3,ngp); sig=zeros(6,ngp);
for gp=1:ngp
  xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zet=GpLoc(gp,3);
  dNr=dershapefunc2D(xsi,eta,zet);
  dxr=dNr*(coord+uvw(reshape(1:3*nen,3,nen))');
  detJ=det(dxr); dNx=dxr\dNr;
  dF=inv(eye(3)-duvw(reshape(1:3*nen,3,nen))*dNx.');
  F(:,:,gp)=dF*Fold(:,:,gp);
  J=det(F(:,:,gp));
  lamb=lam/J; miub=(miu-lam*log(J))/J;
  B=formB(dNx,nen);
  D=[lamb*ones(3)+2*miub*eye(3) zeros(3); zeros(3) miub*eye(3)];
  s=(miu/J)*(F(:,:,gp)*F(:,:,gp)'-eye(3)) + (lam/J)*log(J)*eye(3);
  for n=1:nen
    for m=1:nen
      Ks(n*3-2:n*3,m*3-2:m*3)=dNx(:,n)'*s*dNx(:,m)*eye(3);
    end
  end
  sig(:,gp)=s([1,5,9,2,6,3])';
  ke=ke+(B'*D*B+Ks)*detJ*wp(gp);
  fint=fint+B.'*sig(:,gp)*detJ*wp(gp);
end
```

Listing 3.5: Function file required for the *Updated Lagrangian* analysis.

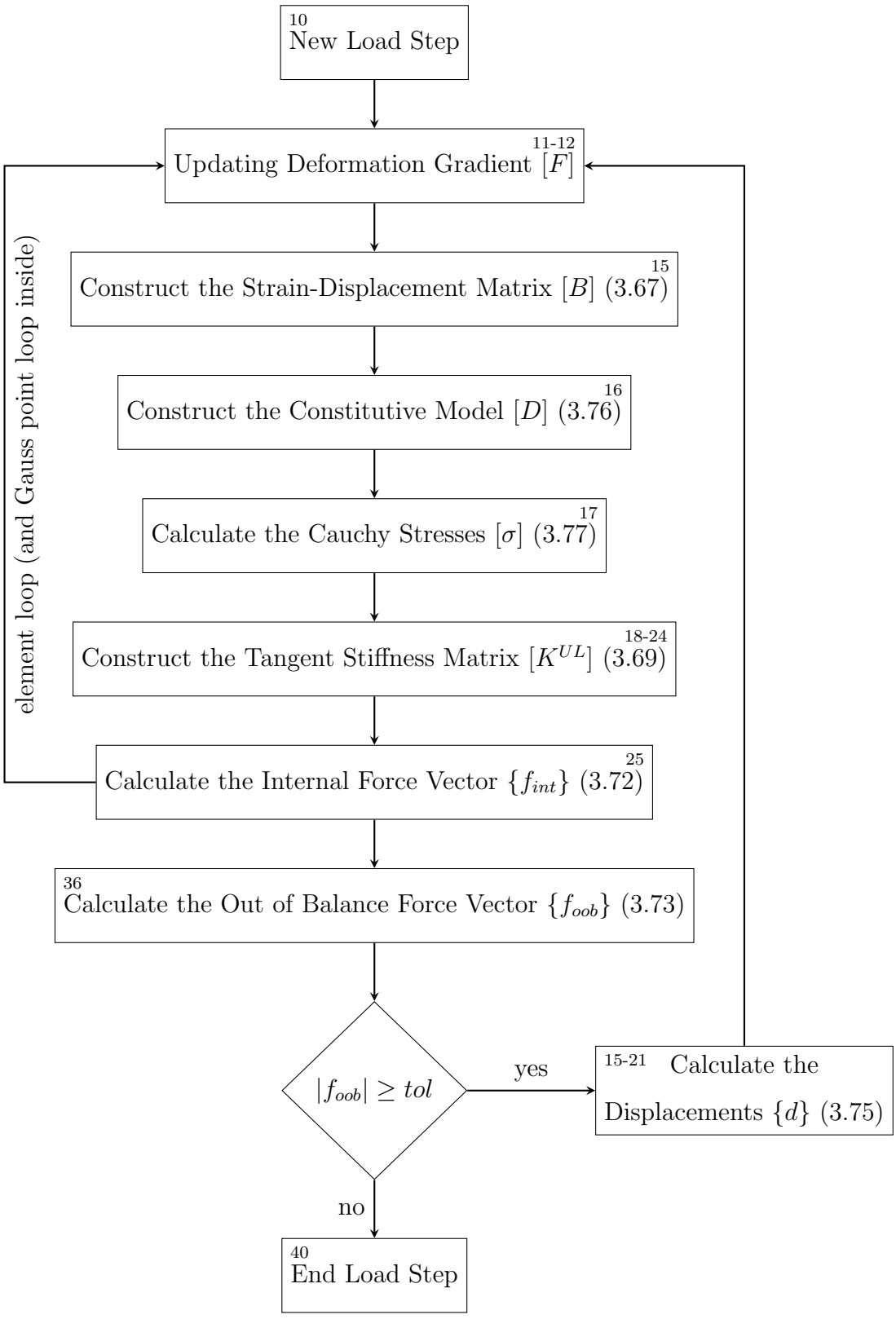


Figure 3.10: Flow diagram depicting the sequential operation within a single load step for the *Updated Lagrangian* FE analysis for a *Neo-Hookean* material model.

The data input format for Listing 3.4 is given in the Table below.

Variable	Matrix Sizing	Format
`coord`	$nen \times 3$	$\begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}$ (in m)
`etpl`	$nels \times nen$	$\begin{bmatrix} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 & n_7 & n_8 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$
`fext0`	$(nen \times dof) \times 1$	$\begin{Bmatrix} fext0_1 \\ fext0_2 \\ \vdots \end{Bmatrix}$ (in N)
`bc`	$nbc \times 2$	$\begin{bmatrix} node\_number & node\_displacement \\ \vdots & \vdots \end{bmatrix}$
`lam`	Scalar	λ (in Pa)
`miu`	Scalar	μ (in Pa)
`ngp`	Scalar	ngp
`lstps`	Scalar	$lstps$
`NRitmax`	Scalar	$NRitmax$
`NRtol`	Scalar	$NRtol$

Table 3.9: Input format for the 8-noded hexahedral analysis for material non-linear case, with example values for `lam`, `miu`, `ngp`, `lstps`, `NRitmax` and `NRtol`.

The output format is shown in Table 3.10.

Variable	Matrix Sizing	Format
`uvw`	$(nen \times dof) \times 1$	$\begin{Bmatrix} u_1 \\ v_1 \\ \vdots \end{Bmatrix}$ (in metres)

Table 3.10: Output from the 8-noded hexahedral with non-linear material behaviour analysis.

3.2.4 Incompressibility

Most materials will undergo a volume change when exposed to an external force. However, in the special cases where the material undergoes almost no, or very little change in volume (described, for example, by a *Poisson's ratio* approaching 0.5 in an isotropic material) a conventional FE analysis would result in *locking* of the elements and a gross under-prediction of the shear deformations. Such cases are known as *near incompressible* or *incompressible* behaviour. They require special treatment when undertaking FE analysis.

One approach to overcome element locking is referred to as the *mean dilation method*[9]. This approach was adopted due its simplicity (require lesser computational resources compare to other methods) (Scovazzi et al.; 2015). This remedy replaces the volumetric strain at the *Gaussian integration points* with the average element volumetric strain.

To ensure that the material remains incompressible, the stiffness matrix has to be modified (expanded) to satisfy the conservation of volume shown below

$$\left[K^{UL}\right] = \left[K_L^{UL}\right] + \left[K_{NL}^{UL}\right] + \left[K_V^{UL}\right] \tag{3.78}$$

where $[K_V^{UL}]$ is the volume conserving stiffness matrix. The linear components of the stiffness matrix are expressed as follows

$$\left[K_L^{UL}\right] = \left[G\right]^T \left[D\right] \left[G\right] \tag{3.79}$$

where

$$\left[G\right] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_2}{\partial x} & 0 & 0 & \dots & \frac{\partial N_8}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & \dots & 0 & \frac{\partial N_8}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & \dots & 0 & 0 & \frac{\partial N_8}{\partial z} \\ \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & 0 & \dots & \frac{\partial N_8}{\partial y} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_2}{\partial x} & 0 & \dots & 0 & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & 0 & \dots & 0 & \frac{\partial N_8}{\partial z} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & \dots & 0 & 0 & \frac{\partial N_8}{\partial y} \\ 0 & 0 & \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_2}{\partial x} & \dots & 0 & 0 & \frac{\partial N_8}{\partial x} \\ \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & 0 & 0 & \dots & \frac{\partial N_8}{\partial z} & 0 & 0 \end{bmatrix} \tag{3.80}$$

and

[9]This approach is also known as the *Selective Reduced Integration*, *B-Bar* and *Constant Volume* Methods.

$$\begin{aligned}\left[D\right] =& 2\mu J^{-\frac{5}{3}}\left(\tfrac{1}{6}I_b\left(\left[I_9\right]+\left[I_{93}\right]\right)-\tfrac{1}{3}\left[I_{9b}\right]-\tfrac{1}{3}\left[I_{9b}\right]^T+\tfrac{1}{9}\left[I_{39}\right]\right)\\ &+p\left(\left[I_{39}\right]-\left[I_9\right]-\left[I_{93}\right]\right)\end{aligned} \tag{3.81}$$

The component matrices $[I_9]$, $[I_{9b}]$, $[I_{93}]$ and $[I_{39}]$ are used in (3.81) to allow the incompressible material model to be described in matrix form rather than the tensorial notation supplied by Bonet and Wood (Bonet and Wood; 1997). They are given as follows

$$\left[I_9\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \left[I_{9b}\right] = \begin{bmatrix} b_{11} & b_{11} & b_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{22} & b_{22} & b_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{33} & b_{33} & b_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{21} & b_{21} & b_{21} & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{12} & b_{12} & b_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{32} & b_{32} & b_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{23} & b_{23} & b_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{13} & b_{13} & b_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{31} & b_{31} & b_{31} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.82}$$

$$\left[I_{93}\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \left[I_{39}\right] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.83}$$

Note that b_{11} refers to a component of the *left Cauchy-Green matrix* of (3.41) as illustrated below

$$\left[b\right] = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \tag{3.84}$$

The non-linear components of the stiffness matrix remain largely the same as (3.70), except that the *Cauchy* stresses are adjusted to account for the incompressiblity (refer to Section 5.5.2 of (Bonet and Wood; 1997) for the process). The final expression for the Cauchy stresses is as follows

$$\left[\sigma\right] = \mu J^{-\frac{5}{3}} \left(\left[b\right] - \tfrac{1}{3} I_b \left[I\right] \right) + p \left[I\right] \tag{3.85}$$

where μ and J are as given after (3.76),

$$I_b = trace \left[b\right] \tag{3.86}$$

and

$$p = \kappa \left(\bar{J} - 1 \right) \tag{3.87}$$

In (3.87), κ is the bulk modulus and $\bar{J}$ is the ratio of the deformed volume, V_d, to the initial volume, V

$$\bar{J} = \frac{V_d}{V} \tag{3.88}$$

where

$$V_d = \sum_{i=1}^{GP} det \left[J_d\right] \omega_i \tag{3.89}$$

ω_i is the weighting of a *Gaussian integration point* and $[J_d]$ is the Jacobian based on the deformed configuration as shown below

$$\left[J_d\right] = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \dots & \frac{\partial N_8}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \dots & \frac{\partial N_8}{\partial \eta} \\ \frac{\partial N_1}{\partial \zeta} & \frac{\partial N_2}{\partial \zeta} & \dots & \frac{\partial N_8}{\partial \zeta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_8 & y_8 & z_8 \end{bmatrix} \tag{3.90}$$

$$V = \sum_{i=1}^{GP} det\left[J\right] \omega_i \tag{3.91}$$

The volume-conserving stiffness matrix is expressed as

$$\left[K_V^{UL}\right] = \begin{bmatrix} K_{V_{11}}^{UL} & K_{V_{12}}^{UL} & \dots & K_{V_{1n}}^{UL} \\ K_{V_{21}}^{UL} & \ddots & & \\ \vdots & & & \\ K_{V_{n1}}^{UL} & \dots & & K_{V_{nn}}^{UL} \end{bmatrix} \tag{3.92}$$

where, for example,

$$\begin{aligned} K_{V_{11}}^{UL} &= \bar{\kappa} v \begin{Bmatrix} \frac{\partial \bar{N}_1}{\partial x} \\ \frac{\partial \bar{N}_1}{\partial y} \\ \frac{\partial \bar{N}_1}{\partial z} \end{Bmatrix} \begin{Bmatrix} \frac{\partial \bar{N}_1}{\partial x} \\ \frac{\partial \bar{N}_1}{\partial y} \\ \frac{\partial \bar{N}_1}{\partial z} \end{Bmatrix}^T \\ K_{V_{12}}^{UL} &= \bar{\kappa} v \begin{Bmatrix} \frac{\partial \bar{N}_1}{\partial x} \\ \frac{\partial \bar{N}_1}{\partial y} \\ \frac{\partial \bar{N}_1}{\partial z} \end{Bmatrix} \begin{Bmatrix} \frac{\partial \bar{N}_2}{\partial x} \\ \frac{\partial \bar{N}_2}{\partial y} \\ \frac{\partial \bar{N}_2}{\partial z} \end{Bmatrix}^T \end{aligned} \tag{3.93}$$

and

$$\begin{Bmatrix} \frac{\partial \bar{N}_1}{\partial x} \\ \frac{\partial \bar{N}_1}{\partial y} \\ \frac{\partial \bar{N}_1}{\partial z} \end{Bmatrix} = \frac{1}{v} \sum_{i=1}^{GP} \begin{Bmatrix} \frac{\partial N_{1_i}}{\partial x} \\ \frac{\partial N_{1_i}}{\partial y} \\ \frac{\partial N_{1_i}}{\partial z} \end{Bmatrix} det\left[J_d\right] \omega_i \tag{3.94}$$

$$\bar{\kappa} = \kappa \bar{J} \tag{3.95}$$

It is noted that the *mean shape functions*, $\bar{N}$, used in the volume conserving stiffness matrix are the mean of all the shape function values at all of the *Gaussian integration points* within an element. The volume-conserving stiffness matrix is constructed after the linear and non-linear stiffness matrices have already been formed at each of the *Gaussian integration points*.

Procedure of a FE analysis incorporating the mean dilation approach

The mean dilation approach does not require any changes to the *Updated Lagrangian* procedure, so Listing 3.4 can be used. However, the function `ULFE` called on line 25-27 of Listing 3.4 has to be replaced by `UNIFE` in order to invoke the mean dilation approach. The mean dilation functions are shown in Listing 3.6. Note that the input and output format will be as listed in Table 3.9 and 3.10 respectively as the same main file was used.

```
function[ke,fint,epsE,sig,F]=...
  ULIFE(nodeData,K,miu,uvw,duvw,ngp,Fold)
nen=size(nodeData,1); coord=nodeData(:,1:3); epsE=zeros(6,ngp);
[wp,GpLoc]=GpPos(ngp); ke=zeros(nen*3); Ks=ke; Kp=ke;
fint=zeros(nen*3,1); F=zeros(3,3,ngp); sig=zeros(6,ngp);
I39=[ones(3,3) zeros(3,6); zeros(6,9)];
I9=eye(9);
I93=zeros(9); I93([1, 11, 21, 32, 40, 52, 60, 72, 80])=1;
ve=0; Ve=0; dNxm=zeros(3,nen);
for gp=1:ngp
  xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zet=GpLoc(gp,3);
  dNr=dershapefunc2D(xsi,eta,zet);
  dXr=dNr*coord; dxr=dNr*(coord+uvw(reshape(1:3*nen,3,nen))');
  dNx=dxr\dNr; detJ0=det(dXr); detJ=det(dxr);
  JW=detJ*wp(gp); ve=ve+JW;
  dNxm=dNxm+dNx*JW;
  Ve=Ve+detJ0*wp(gp);
end
dNxm=dNxm/ve; Jbar=ve/Ve;
p=K*(Jbar-1); Kbar=K*Jbar;
for gp=1:ngp
  xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zet=GpLoc(gp,3);
  dNr=dershapefunc2D(xsi,eta,zet);
  dxr=dNr*(coord+uvw(reshape(1:3*nen,3,nen))');
  detJ=det(dxr); dNx=dxr\dNr;
  dF=inv(eye(3)-duvw(reshape(1:3*nen,3,nen))*dNx.');
  F(:,:,gp)=dF*Fold(:,:,gp);
  J=det(F(:,:,gp)); b=F(:,:,gp)*F(:,:,gp)';
  Ib=trace(b); b9=b([1,5,9,4,2,8,6,3,7])';
  I9b1=[b9 b9 b9 zeros(9,6)]; I9b2=[b9'; b9'; b9'; zeros(6,9)];
  [B,G]=formBG(dNx,nen);
  s=(miu*J^(-5/3))*(b - 1/3*Ib*eye(3)); s=s+p*eye(3);
  for n=1:nen
    for m=1:nen
      Ks(n*3-2:n*3,m*3-2:m*3)=dNx(:,n)'*s*dNx(:,m)*eye(3);
    end
  end
  sig(:,gp)=s([1,5,9,2,6,3])';
  D=2*(miu*J^(-5/3))*...
      ((1/6)*Ib*(I9+I93)-(1/3)*I9b1-(1/3)*I9b2+(1/9)*Ib*I39);
  D=D+p*(I39-I9-I93);
  ke=ke+(G'*D*G+Ks)*detJ*wp(gp);
  fint=fint+B.'*sig(:,gp)*detJ*wp(gp);
end
for n=1:nen
  for m=1:nen
    Kp(n*3-2:n*3,m*3-2:m*3)=Kbar*ve*(dNxm(:,n)*dNxm(:,m)');
  end
end
ke=ke+Kp;
```

```
51
52 function [B,G]=formBG(dNx,nen)
53 B=zeros(6,nen*3);
54 B(1,1:3:end)=dNx(1,:); B(2,2:3:end)=dNx(2,:);
55 B(3,3:3:end)=dNx(3,:); B(4,1:3:end)=dNx(2,:);
56 B(4,2:3:end)=dNx(1,:); B(5,2:3:end)=dNx(3,:);
57 B(5,3:3:end)=dNx(2,:); B(6,1:3:end)=dNx(3,:);
58 B(6,3:3:end)=dNx(1,:);
59 G=zeros(9,nen*3); G(1:3,:)=B(1:3,:);
60 G(4,1:3:end)=dNx(2,:); G(5,2:3:end)=dNx(1,:);
61 G(6,2:3:end)=dNx(3,:); G(7,3:3:end)=dNx(2,:);
62 G(8,3:3:end)=dNx(1,:); G(9,1:3:end)=dNx(3,:);
```

Listing 3.6: Function files required for an *Updated Lagrangian* analysis using the Mean-Dilation approach.

The number in the top right of the box in Figure 3.11 refers to Listing 3.6 while the number on the top left refers to Listing 3.4.

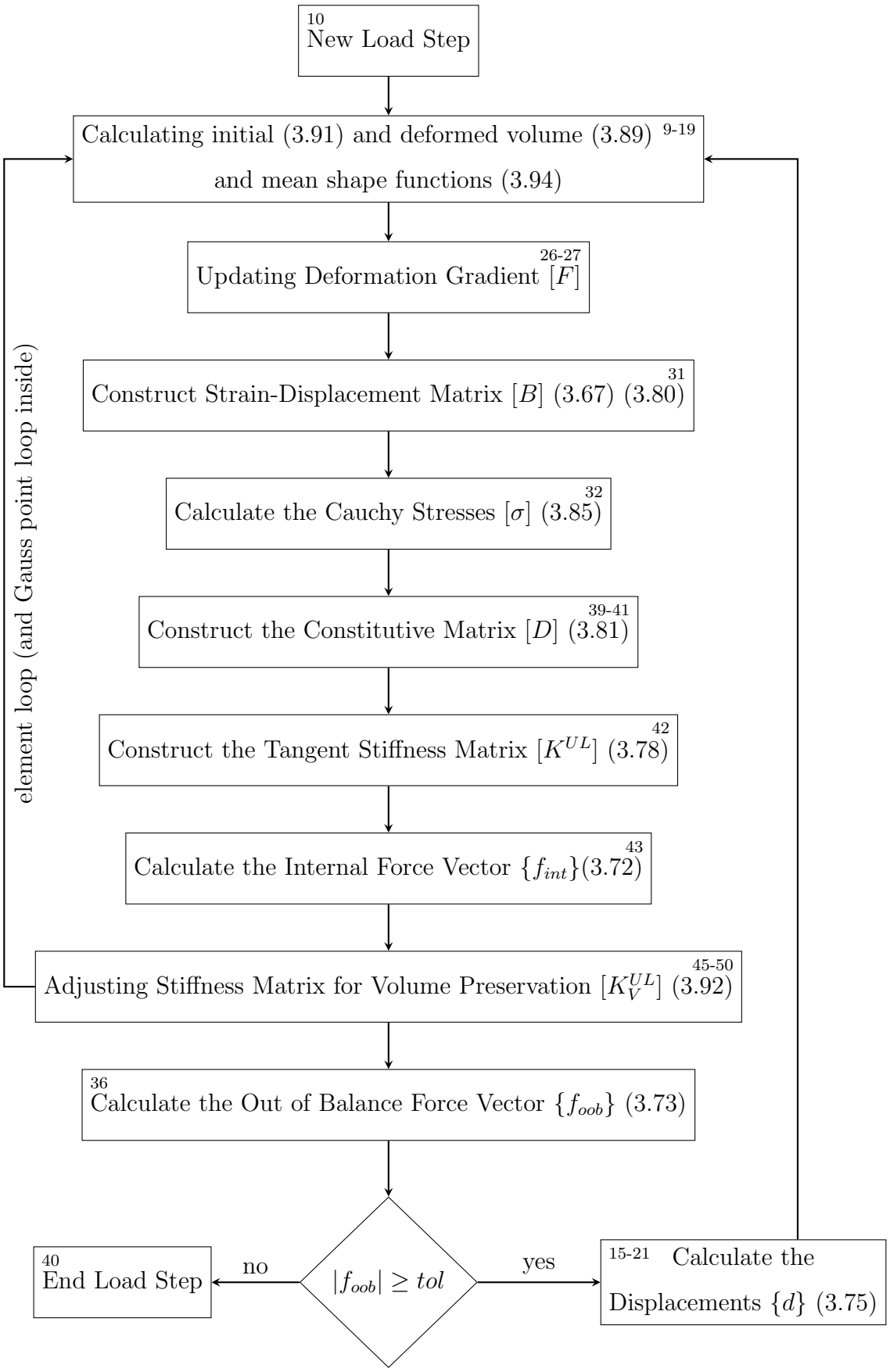


Figure 3.11: Flow diagram depicting the sequential operation within a single load step for the *Updated Lagrangian* FE analysis for an incompressible *Neo-Hookean* material model.

Incompressible Neo-Hookean FE numerical example

Cook's membrane problem offers a useful benchmark test to explore whether an *Updated Lagrangian* FE analysis is capable of reproducing incompressible, or near incompressible large deformation behaviour. The structure is a skewed deep cantilever, subjected to end shear. In this analysis, the particular values for κ and μ have been chosen to replicate those values used by (Simo and Armero; 1992) as shown in Figure 3.12, with its parameters summarised in Table 3.11. While this is a 2D plane strain problem, the 3D hexahedral FE code was used to undertake the simulation (enforcing out-of-plane displacements to be zero).

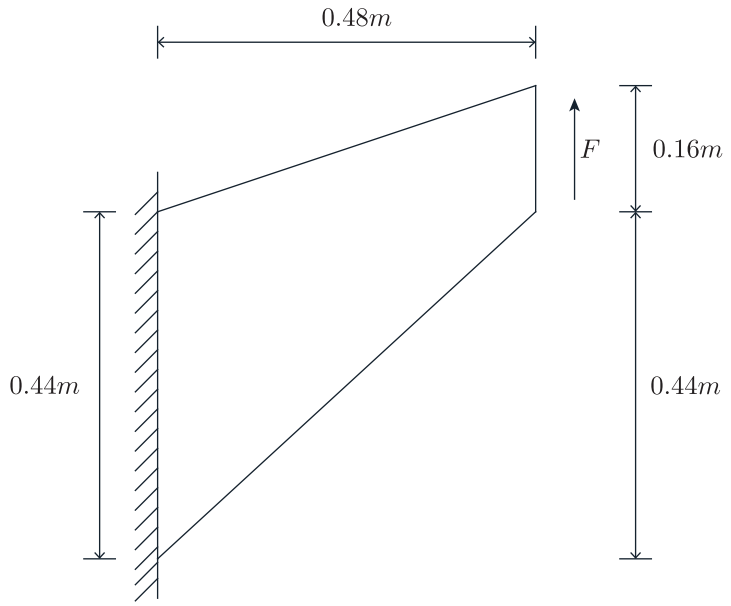
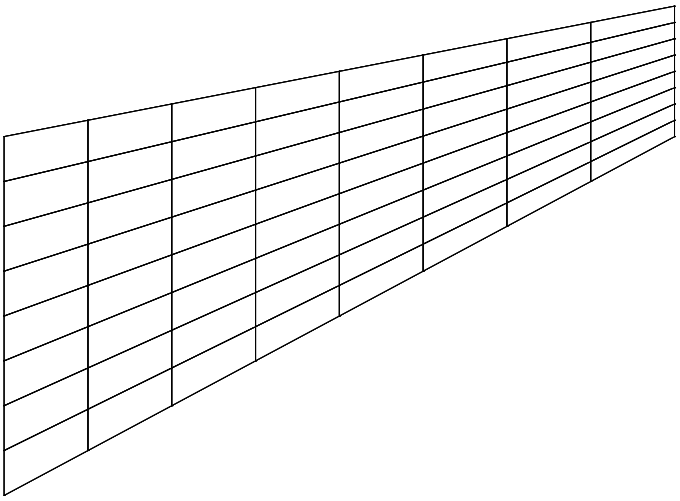


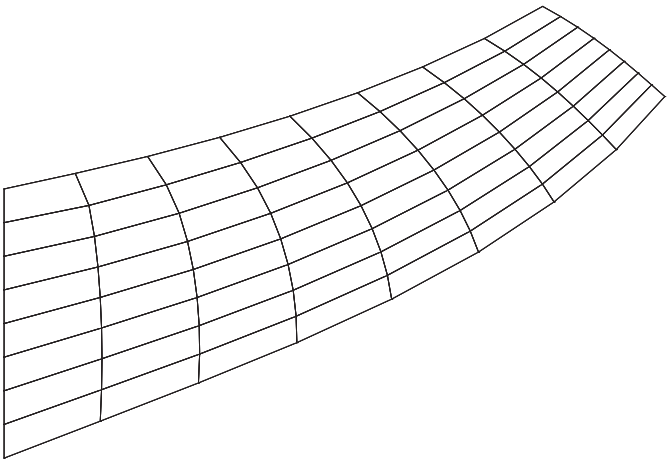
Figure 3.12: The geometry of the Cook's membrane problem.

Bulk modulus, κ	400.942×10^3 Pa
Shear modulus, μ	80.1938 Pa
Force applied, F	100 N

Table 3.11: Input parameters for the incompressible Cook's membrane problem.



(a) Mesh used for Cook's membrane problem.



(b) Deformed mesh in Cook's membrane problem.

Figure 3.13: The undeformed and deformed meshes for the Cook's membrane's problem.

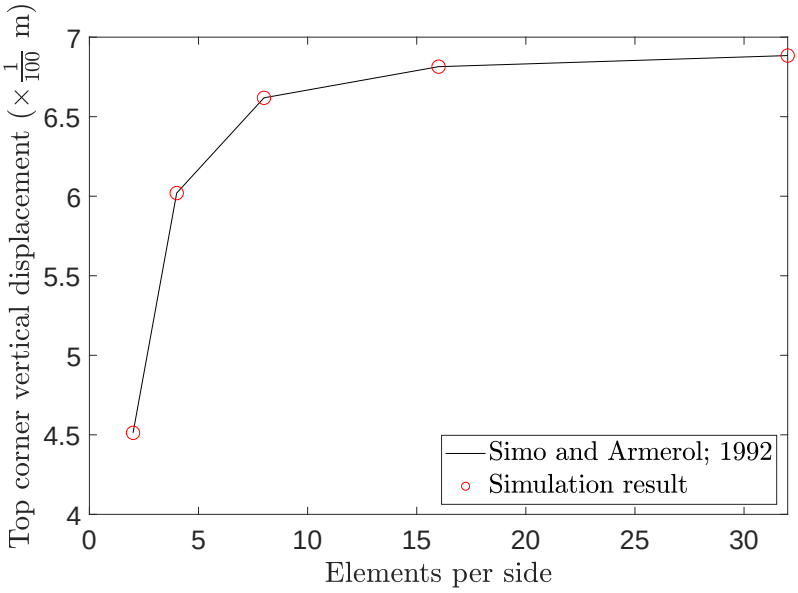


Figure 3.14: End deformation in Cook's membrane problem compared using different mesh densities using 8-noded hexahedral elements. Note that all measures on the ordinate should be divided by 100 to provide the displacement in m.

It can be seen from the results obtained, that the *mean dilation method* has allowed the near incompressible Cook's membrane problem analysed to behave in agreement with those obtained by Simo and Armero (Simo and Armero; 1992). It gives confident that the technique is capable of resolving locking issue that would otherwise dominate this problem.

In this chapter, the FE formulations were introduced and benchmark problems used to provide confidence in the robustness and accuracy of the algorithms. The geometrically non-linear deformation is crucial in modelling the cardiac muscles due to the huge strains experienced. Although material non-linearity and incompressibility were briefly touched on in this chapter, they will not be used in the later simulations due to the nature of the *active strain* approach (described in Section 5.3.2). All subsequent analyses make use of a *Total Lagrangian* formulation.

References

Bathe, K.-J. (2014). *Finite Element Procedures*, second edn, Prentice Hall, Pearson Education, Inc.

Bonet, J. and Wood, R. D. (1997). *Nonlinear Continuum Mechanics for Finite Element Analysis*, Cambridge University Press.

Coombs, W., Crouch, R. and Augarde, C. (2010). 70-line 3d finite deformation elastoplastic finite-element code, *Proceedings of the Seventh European Conference on Numerical Methods in Geotechnical Engineering* pp. 151–156.

Felippa, C. and Haugen, B. (2005). A unified formulation of small-strain corotational finite elements: I. theory, *Computer Methods in Applied Mechanics and Engineering* **194(21-24)**: 2285–2335.

Grosberg, A. (2008). *A Bio-Inspired Computational Model of Cardiac Mechanics: Pathology and Development*, PhD thesis, California Institute of Technology.

Jensen, L. R., Rauhe, J. C. and Stegmann, J. (2001). *Finite Elements for Geometric Non-Linear Analysis of Composite Laminates and Sandwich Structures*, PhD thesis, Aalborg University.

Ottosen, N. S. and Petersson, H. (1992). *Introduction to the Finite Element Method*, Prentice Hall International.

Rice, J. (2017). Introduction to finite element methods (asen 5007), `http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/`.

Scovazzi, G., Carnes, B. and Zeng, X. (2015). A simple, stable and accurate tetrahedral finite element for transient, nearly incompressible, linear and nonlinear elasticity: A dynamic variational multiscale approach, *International Journal for Numerical Methods in Engineering* **106(10)**.

Simo, J. and Armero, F. (1992). Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes*, *International Journal for Numerical Methods in Engineering* **33**: 1413–1449.

Sze, K., Liu, X. and Lo, S. (2004). Popular benchmark problems for geometric nonlinear analysis of shells, *Finite Elements in Analysis and Design* **40(11)**: 1551–1569.

4

Shell finite element formulation

Shell Finite Elements have been developed by structural engineers for curved thin surfaces where in-plane stresses dominate. Such elements have nodes at the mid-surface, rather than at an apex identifying the extremities of the upper and lower surfaces. This gives rise to a reduction in the number of degrees of freedom in comparison with hexahedral elements, but more importantly the shell elements are constructed to reproduce bending behaviour without the necessity for more than one element over the thickness direction (normal to the mid-surface). It is recognised that early thin shell elements were more prone to shear and membrane locking than tetrahedral or hexahedral elements due to the assumption about the strain distribution through the thickness. Many approaches have been introduced to try to overcome those problems (for example, see (Laulusa et al.; 2006)).

There are essentially two types of shell elements: (i) conventional displacement and rotation-based (degenerated) shell elements and (ii) purely displacement-based shell elements. The former type of shell elements were first proposed by (Ahmad; 1969) and these have since been widely used. This approach was derived by assigning a mid surface along the element thickness and referring all the variables to it. This leads to a need to incorporate rotational degrees of freedoms so that the thickness of the element could be "recovered". The second approach does not make use of the mid-surface and instead includes variables at the top and bottom of the element just like a solid 3D element. However, the integration for the element is reduced

to two-dimensional, similar to the first approach, with added assumptions for the behaviour across the thickness direction (Parisch; 1995).

The main advantages of the second approach are that it can cope better with contact problems and the coupling between other solid elements is more straight forward, as these elements make use of just displacement degree of freedoms rather than displacements and rotations. However, it is known that the displacement-based elements cannot capture the bending behaviour as well as the conventional shell elements. In this thesis, conventional shell elements will be used.

4.1 Coordinate systems

To describe the shell element geometry, two coordinate systems and two vector triples (as shown in Figure 4.1) will be used,

- *Global Coordinates* (X, Y, Z or x, y, z). The global coordinates define the nodal coordinates within a Cartesian system. The upper-case letter being used to describe the undeformed element while lower case symbols refer to the deformed element.

- *Local Coordinates* (ξ, η, ζ). As in hexahedral elements, these coordinates define the space within the element, ranging (in each local direction) from -1 to 1 across the element.

- *Local Direction Vectors* ($\{e_x\}$, $\{e_y\}$, $\{e_z\}$). These are parallel to the global Cartesian coordinate system but are situated at each node of interest.

- *Thickness Direction Vectors* ($\{V_n\}$, $\{V_1\}$, $\{V_2\}$). These vectors are used to define the thickness direction of the shell element, $\{V_n\}$ and the two associated orthogonal vectors.

A conventional shell element has 5 degrees of freedom at each node; displacements u, v and w along each of the global axes (x, y and z respectively) and two rotations, ϕ and ψ representing angular measures about the local $\{V_1\}$ and $\{V_2\}$ axes respectively.

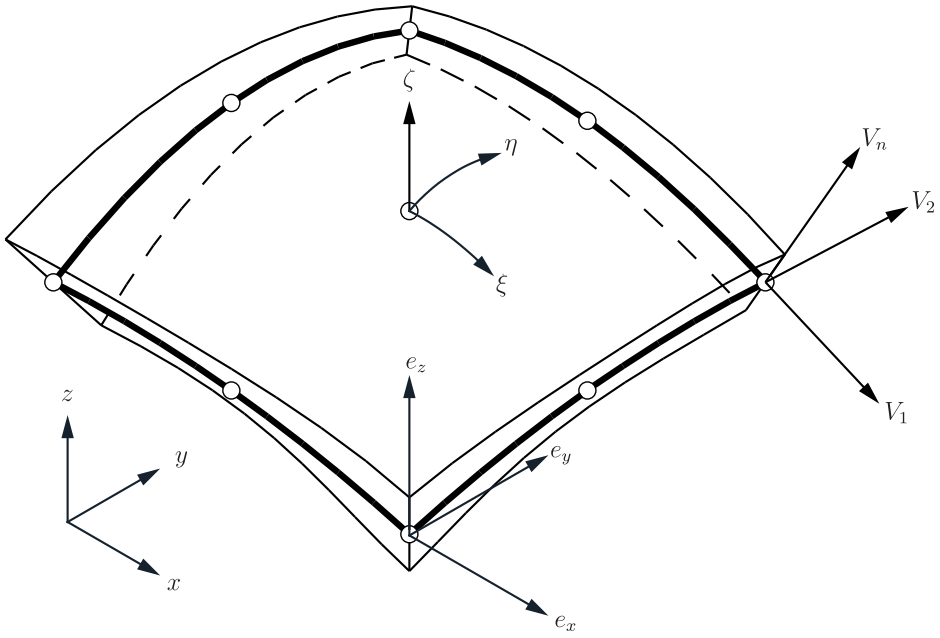


Figure 4.1: Coordinate Systems for a 9-noded shell element.

The initial thickness direction of the shell element, $\{V_n\}$, (a unit vector) is required to be specified at each node. Subsequently, the two unit vectors $\{V_1\}$ and $\{V_2\}$ (both orthogonal to $\{V_n\}$) are defined using (4.1) and (4.2) respectively, where $\{e_y\}$ is the unit vector along the y direction.

$$\Big\{V_1\Big\} = \frac{\Big\{e_y\Big\} \times \Big\{V_n\Big\}}{\Big|\Big\{e_y\Big\}^T \Big\{V_n\Big\}\Big|} \tag{4.1}$$

$$\Big\{V_2\Big\} = \Big\{V_n\Big\} \times \Big\{V_1\Big\} \tag{4.2}$$

where $\times$ denotes the cross product, and $\{e_x\}$ replaces $\{e_y\}$ in (4.1) if $\{V_n\}$ is parallel to the $\{e_y\}$ direction. The thickness vector for the deformed shell could be determined as follows

$$\Big\{V_n\Big\} = \frac{\frac{\partial N}{\partial \xi}\Big\{\Big\{X\Big\} + \Big\{d\Big\}\Big\}}{\Big|\frac{\partial N^i}{\partial \xi}\Big\{\Big\{X\Big\} + \Big\{d\Big\}\Big\}\Big|} \times \frac{\frac{\partial N}{\partial \eta}\Big\{\Big\{X\Big\} + \Big\{d\Big\}\Big\}}{\Big|\frac{\partial N^n}{\partial \eta}\Big\{\Big\{X\Big\} + \Big\{d\Big\}\Big\}\Big|} \tag{4.3}$$

where $\{X\}$ gives the coordinates of the node and $\{d\}$ gives the displacement of the

node. Superscript i refers to the node under consideration. The deformations at any point within the shell element are given by

$$\left\{d\right\} = \sum_{i=1}^{nen} N^i \left\{d^i\right\} + \frac{\zeta}{2} \sum_{n=1}^{nen} h^i N^i \left\{\Delta V_n^i\right\} \tag{4.4}$$

where nen gives the number of nodes per element, $\{d^i\}$ is the deformation at node i, N^i is the *shape function* for that node and h^i is the thickness of the shell at that node. If we assume that the iterative (incremental) change in the rotations about the $\{{}^tV_1\}$ and $\{{}^tV_2\}$ are small (that is $sin(\Delta\phi) \approx \Delta\phi$) (Jensen et al.; 2001), where $\Delta\phi$ is the rotation about $\{{}^tV_1\}$ from load step t to load step $t + \Delta t$, then we can approximate $\{\Delta V_n\}$ by (4.5)

$$\left\{\Delta V_n\right\} = -\left\{{}^tV_2\right\}\Delta\phi + \left\{{}^tV_1\right\}\Delta\psi \tag{4.5}$$

(4.6) is then obtained by substituting (4.5) into (4.4).

$$\left\{d\right\} = \sum_{i=1}^{nen} N^i \left\{d^i\right\} + \frac{\zeta}{2} \sum_{i=1}^{nen} h^i N^i \left\{-\left\{{}^tV_2^i\right\}\Delta\phi^i + \left\{{}^tV_1^i\right\}\Delta\psi^i\right\} \tag{4.6}$$

By differentiating the displacements with respect to the local coordinates, we arrive at the terms required for the strain-displacement matrix (Bathe and Bolourchi; 1980)

$$\begin{Bmatrix} \frac{\partial\{d\}^T}{\partial\xi} \\ \frac{\partial\{d\}^T}{\partial\eta} \\ \frac{\partial\{d\}^T}{\partial\zeta} \end{Bmatrix} = \sum_{i=1}^{nen} \begin{bmatrix} \frac{\partial N^i}{\partial\xi}\{u^i + \phi^i\zeta^t g_{11}^i + \psi^i\zeta^t g_{21}^i & v^i + \phi^i\zeta^t g_{12}^i + \psi^i\zeta^t g_{22}^i \\ \frac{\partial N^i}{\partial\eta}\{u^i + \phi^i\zeta^t g_{11}^i + \psi^i\zeta^t g_{21}^i & v^i + \phi^i\zeta^t g_{12}^i + \psi^i\zeta^t g_{22}^i \\ N^i\{\phi^{it} g_{11}^i + \psi^{it} g_{21}^i & \phi^{it} g_{12}^i + \psi^{it} g_{22}^i\} \\ +w^i + \phi^i\zeta^t g_{13}^i + \psi^i\zeta^t g_{23}^i\} & \\ +w^i + \phi^i\zeta^t g_{13}^i + \psi^i\zeta^t g_{23}^i\} & \\ +\phi^{it} g_{13}^i + \psi^{it} g_{23}^i\} & \end{bmatrix} \tag{4.7}$$

where $\{g_1\}$ and $\{g_2\}$ are given by

$$\left\{ {}^{t}g_1^i \right\} = -\frac{1}{2} h^i \left\{ {}^{t}V_2^i \right\} \tag{4.8}$$

$$\left\{ {}^{t}g_2^i \right\} = \frac{1}{2} h^i \left\{ {}^{t}V_1^i \right\} \tag{4.9}$$

Note that (4.7) involves differentiation with respect to the *local coordinates* yet we require the derivatives with respect to the global coordinates. We make use of the Jacobian transformation to arrive at the appropriate expressions

$$\begin{Bmatrix} \frac{\partial \{u\}^T}{\partial X} \\ \frac{\partial \{u\}^T}{\partial Y} \\ \frac{\partial \{u\}^T}{\partial Z} \end{Bmatrix} = \sum_{i=1}^{nen} \begin{bmatrix} u^i \frac{\partial N^i}{\partial X} + \phi^{it} g_{11}^i {}^{t}G_1^i + \psi^{it} g_{21}^i {}^{t}G_1^i & v^i \frac{\partial N^i}{\partial X} + \phi^{it} g_{12}^i {}^{t}G_1^i + \psi^{it} g_{22}^i {}^{t}G_1^i \\ u^i \frac{\partial N^i}{\partial Y} + \phi^{it} g_{11}^i {}^{t}G_2^i + \psi^{it} g_{21}^i {}^{t}G_2^i & v^i \frac{\partial N^i}{\partial Y} + \phi^{it} g_{12}^i {}^{t}G_2^i + \psi^{it} g_{22}^i {}^{t}G_2^i \\ u^i \frac{\partial N^i}{\partial Z} + \phi^{it} g_{11}^i {}^{t}G_3^i + \psi^{it} g_{21}^i {}^{t}G_3^i & v^i \frac{\partial N^i}{\partial Z} + \phi^{it} g_{12}^i {}^{t}G_3^i + \psi^{it} g_{22}^i {}^{t}G_3^i \end{bmatrix}$$
$$\begin{bmatrix} w^i \frac{\partial N^i}{\partial X} + \phi^{it} g_{13}^i {}^{t}G_1^i + \psi^{it} g_{23}^i {}^{t}G_1^i \\ w^i \frac{\partial N^i}{\partial Y} + \phi^{it} g_{13}^i {}^{t}G_2^i + \psi^{it} g_{23}^i {}^{t}G_2^i \\ w^i \frac{\partial N^i}{\partial Z} + \phi^{it} g_{13}^i {}^{t}G_3^i + \psi^{it} g_{23}^i {}^{t}G_3^i \end{bmatrix} \tag{4.10}$$

where $\frac{\partial N^i}{\partial X}$ and $\{G\}$ are defined by (4.11) and (4.12) respectively, and the Jacobian matrix is given in (4.13).

$$\begin{Bmatrix} \frac{\partial N^i}{\partial X} \\ \frac{\partial N^i}{\partial Y} \\ \frac{\partial N^i}{\partial Z} \end{Bmatrix} = \begin{Bmatrix} (J^{-1})_{11} \frac{\partial N^i}{\partial \xi} + (J^{-1})_{12} \frac{\partial N^i}{\partial \eta} \\ (J^{-1})_{21} \frac{\partial N^i}{\partial \xi} + (J^{-1})_{22} \frac{\partial N^i}{\partial \eta} \\ (J^{-1})_{31} \frac{\partial N^i}{\partial \xi} + (J^{-1})_{32} \frac{\partial N^i}{\partial \eta} \end{Bmatrix} \tag{4.11}$$

$$\left\{ {}^{t}G^i \right\} = h \begin{Bmatrix} \zeta \left((J^{-1})_{11} \frac{\partial N^i}{\partial \xi} + (J^{-1})_{12} \frac{\partial N^i}{\partial \eta} \right) + (J^{-1})_{13} N^i \\ \zeta \left((J^{-1})_{21} \frac{\partial N^i}{\partial \xi} + (J^{-1})_{22} \frac{\partial N^i}{\partial \eta} \right) + (J^{-1})_{23} N^i \\ \zeta \left((J^{-1})_{31} \frac{\partial N^i}{\partial \xi} + (J^{-1})_{32} \frac{\partial N^i}{\partial \eta} \right) + (J^{-1})_{33} N^i \end{Bmatrix} \tag{4.12}$$

$$\begin{aligned}\left[J\right] &= \begin{bmatrix} \frac{\partial X}{\partial \xi} & \frac{\partial Y}{\partial \xi} & \frac{\partial Z}{\partial \xi} \\ \frac{\partial X}{\partial \eta} & \frac{\partial Y}{\partial \eta} & \frac{\partial Z}{\partial \eta} \\ \frac{\partial X}{\partial \zeta} & \frac{\partial Y}{\partial \zeta} & \frac{\partial Z}{\partial \zeta} \end{bmatrix} \\ &= \sum_{i=1}^{nen} \begin{bmatrix} \frac{\partial N^i}{\partial \xi} \left\{ \{x^i\} + 0.5 t^i \zeta \{V_n^i\} \right\}^T \\ \frac{\partial N^i}{\partial \eta} \left\{ \{x^i\} + 0.5 t^i \zeta \{V_n^i\} \right\}^T \\ N^i \left\{ 0.5 h^i \{V_n^i\} \right\}^T \end{bmatrix} \end{aligned} \tag{4.13}$$

where

$$\left[J\right]^{-1} = \begin{bmatrix} (J^{-1})_{11} & (J^{-1})_{12} & (J^{-1})_{13} \\ (J^{-1})_{21} & (J^{-1})_{22} & (J^{-1})_{23} \\ (J^{-1})_{31} & (J^{-1})_{32} & (J^{-1})_{33} \end{bmatrix} \tag{4.14}$$

Note that $(J^{-1})_{12}$ refers to the row 1 and column 2 component of the inverse of the Jacobian (the Jacobian matrix have to be inverse as a whole and then divided into those components). Use of (3.6) from the hexahedral finite element is no longer valid. To incorporate the rotation degrees of freedoms into the $[B_s^{TL}]$ matrix requires expansion as follows

$$\left[B_s^{TL}\right] = \left[B_{sL}^{TL}\right] + \left[B_{sNL}^{TL}\right] \tag{4.15}$$

where the linear components, $[B_{sL}^{TL}]$ are given as (Jensen et al.; 2001)(Bathe and Bolourchi; 1980)

$$\overbrace{\left[B_{sL}^{TL}\right]}^{6\times 5n} = \begin{bmatrix} \dots & \frac{\partial N^i}{\partial X} & 0 & 0 & g_{11}^i G_1^i & g_{21}^i G_1^i & \dots \\ \dots & 0 & \frac{\partial N^i}{\partial Y} & 0 & g_{12}^i G_2^i & g_{22}^i G_2^i & \dots \\ \dots & 0 & 0 & \frac{\partial N^i}{\partial Z} & g_{13}^i G_3^i & g_{23}^i G_3^i & \dots \\ \dots & \frac{\partial N^i}{\partial Y} & \frac{\partial N^i}{\partial X} & 0 & g_{11}^i G_2^i + g_{12}^i G_1^i & g_{21}^i G_1^i + g_{22}^i G_1^i & \dots \\ \dots & \frac{\partial N^i}{\partial Z} & 0 & \frac{\partial N^i}{\partial X} & g_{11}^i G_3^i + g_{13}^i G_1^i & g_{21}^i G_3^i + g_{23}^i G_1^i & \dots \\ \dots & 0 & \frac{\partial N^i}{\partial Z} & \frac{\partial N^i}{\partial Y} & g_{12}^i G_3^i + g_{13}^i G_2^i & g_{22}^i G_3^i + g_{23}^i G_2^i & \dots \end{bmatrix} \tag{4.16}$$

The non-linear components, $[B_{sNL}^{TL}]$ are given as follows

$$\overbrace{\left[B_{sNL}^{TL}\right]}^{6\times 5n} = \begin{bmatrix} \dots & \frac{\partial N^i}{\partial X}\frac{\partial u^i}{\partial X} & \frac{\partial N^i}{\partial X}\frac{\partial v^i}{\partial X} & \frac{\partial N^i}{\partial X}\frac{\partial w^i}{\partial X} \\ \dots & \frac{\partial N^i}{\partial Y}\frac{\partial u^i}{\partial Y} & \frac{\partial N^i}{\partial Y}\frac{\partial v^i}{\partial Y} & \frac{\partial N^i}{\partial Y}\frac{\partial w^i}{\partial Y} \\ \dots & \frac{\partial N^i}{\partial Z}\frac{\partial u^i}{\partial Z} & \frac{\partial N^i}{\partial Z}\frac{\partial v^i}{\partial Z} & \frac{\partial N^i}{\partial Z}\frac{\partial w^i}{\partial Z} \\ \dots & \frac{\partial N^i}{\partial Y}\frac{\partial u^i}{\partial X} + \frac{\partial N^i}{\partial X}\frac{\partial u^i}{\partial Y} & \frac{\partial N^i}{\partial Y}\frac{\partial v^i}{\partial X} + \frac{\partial N^i}{\partial X}\frac{\partial v^i}{\partial Y} & \frac{\partial N^i}{\partial Y}\frac{\partial w^i}{\partial X} + \frac{\partial N^i}{\partial X}\frac{\partial w^i}{\partial Y} \\ \dots & \frac{\partial N^i}{\partial Z}\frac{\partial u^i}{\partial Y} + \frac{\partial N^i}{\partial Y}\frac{\partial u^i}{\partial Z} & \frac{\partial N^i}{\partial Z}\frac{\partial v^i}{\partial Y} + \frac{\partial N^i}{\partial Y}\frac{\partial v^i}{\partial Z} & \frac{\partial N^i}{\partial Z}\frac{\partial w^i}{\partial Y} + \frac{\partial N^i}{\partial Y}\frac{\partial w^i}{\partial Z} \\ \dots & \frac{\partial N^i}{\partial Z}\frac{\partial u^i}{\partial X} + \frac{\partial N^i}{\partial X}\frac{\partial u^i}{\partial Z} & \frac{\partial N^i}{\partial Z}\frac{\partial v^i}{\partial X} + \frac{\partial N^i}{\partial X}\frac{\partial v^i}{\partial Z} & \frac{\partial N^i}{\partial Z}\frac{\partial w^i}{\partial X} + \frac{\partial N^i}{\partial X}\frac{\partial w^i}{\partial Z} \end{bmatrix} \tag{4.17}$$

$$\begin{bmatrix} \chi_{11}^i G_1^i & \chi_{21}^i G_1^i & \dots \\ \chi_{12}^i G_2^i & \chi_{22}^i G_2^i & \dots \\ \chi_{13}^i G_3^i & \chi_{23}^i G_3^i & \dots \\ \chi_{12}^i G_1^i + \chi_{11}^i G_2^i & \chi_{22}^i G_1^i + \chi_{21}^i G_2^i & \dots \\ \chi_{13}^i G_2^i + \chi_{12}^i G_3^i & \chi_{23}^i G_2^i + \chi_{22}^i G_3^i & \dots \\ \chi_{13}^i G_1^i + \chi_{11}^i G_3^i & \chi_{23}^i G_1^i + \chi_{21}^i G_3^i & \dots \end{bmatrix}$$

where

$$
\begin{aligned}
\chi^i_{11} &= g^i_{11}\frac{\partial u^i}{\partial X^i} + g^i_{12}\frac{\partial v^i}{\partial X^i} + g^i_{13}\frac{\partial w^i}{\partial X^i} \\
\chi^i_{12} &= g^i_{11}\frac{\partial u^i}{\partial Y^i} + g^i_{12}\frac{\partial v^i}{\partial Y^i} + g^i_{13}\frac{\partial w^i}{\partial Y^i} \\
\chi^i_{13} &= g^i_{11}\frac{\partial u^i}{\partial Z^i} + g^i_{12}\frac{\partial v^i}{\partial Z^i} + g^i_{13}\frac{\partial w^i}{\partial Z^i} \\
\chi^i_{21} &= g^i_{21}\frac{\partial u^i}{\partial X^i} + g^i_{22}\frac{\partial v^i}{\partial X^i} + g^i_{23}\frac{\partial w^i}{\partial X^i} \\
\chi^i_{22} &= g^i_{21}\frac{\partial u^i}{\partial Y^i} + g^i_{22}\frac{\partial v^i}{\partial Y^i} + g^i_{23}\frac{\partial w^i}{\partial Y^i} \\
\chi^i_{23} &= g^i_{21}\frac{\partial u^i}{\partial Z^i} + g^i_{22}\frac{\partial v^i}{\partial Z^i} + g^i_{23}\frac{\partial w^i}{\partial Z^i}
\end{aligned}
\tag{4.18}
$$

These non-linear components can be formed similarly to (3.64). By making use of (3.61), (4.17) can be re-written as

$$
\left[B^{TL}_{sNL}\right] = \left[A\right]\left[G\right] \tag{4.19}
$$

where

$$
\overbrace{\left[G\right]}^{9\times 5n} = \begin{bmatrix}
\dots & \frac{\partial N^i}{\partial X} & 0 & 0 & g^i_{11}G^i_1 & g^i_{21}G^i_1 & \dots \\
\dots & 0 & \frac{\partial N^i}{\partial X} & 0 & g^i_{12}G^i_1 & g^i_{22}G^i_1 & \dots \\
\dots & 0 & 0 & \frac{\partial N^i}{\partial X} & g^i_{13}G^i_1 & g^i_{23}G^i_1 & \dots \\
\dots & \frac{\partial N^i}{\partial Y} & 0 & 0 & g^i_{11}G^i_2 & g^i_{21}G^i_2 & \dots \\
\dots & 0 & \frac{\partial N^i}{\partial Y} & 0 & g^i_{12}G^i_2 & g^i_{22}G^i_2 & \dots \\
\dots & 0 & 0 & \frac{\partial N^i}{\partial Y} & g^i_{13}G^i_2 & g^i_{23}G^i_2 & \dots \\
\dots & \frac{\partial N^i}{\partial Z} & 0 & 0 & g^i_{11}G^i_3 & g^i_{21}G^i_3 & \dots \\
\dots & 0 & \frac{\partial N^i}{\partial Z} & 0 & g^i_{12}G^i_3 & g^i_{22}G^i_3 & \dots \\
\dots & 0 & 0 & \frac{\partial N^i}{\partial Z} & g^i_{13}G^i_3 & g^i_{23}G^i_3 & \dots
\end{bmatrix}
\tag{4.20}
$$

Now that both the linear and non-linear strain-displacement matrices have been defined, the tangent stiffness matrix, $[K^{TL}_s]$ can then be determined using the following expression

$$
\left[K^{TL}_s\right] = \left[K^{TL}_{sL}\right] + \left[K^{TL}_{sNL}\right] \tag{4.21}
$$

where $[K_{sL}^{TL}]$ is the linear stiffness matrix and $[K_{sNL}^{TL}]$ is the non-linear stiffness matrix. The linear stiffness matrix is given as follows

$$\begin{aligned}\Big[K_{sL}^{TL}\Big] &= \int_V [B_s^{TL}]^T[D][B_s^{TL}]dV \\ &= \int_V \Big([B_{sL}^{TL}]^T + [B_{sNL}^{TL}]^T\Big)[D]\Big([B_{sL}^{TL}] + [B_{sNL}^{TL}]\Big)dV\end{aligned} \tag{4.22}$$

where $[D]$ is the constitutive matrix. Equation (4.22) is integrated numerically as

$$\Big[K_{sL}^{TL}\Big] = \sum_{i=1}^{nGP}[B_s^{TL}]^T[D][B_s^{TL}]det\,\Big[J\Big]\,\omega_i \tag{4.23}$$

where ω_i are the Gauss point weights. The non-linear stiffness matrix is given by (Jensen et al.; 2001)

$$\Big[K_{sNL}^{TL}\Big] = \int_V \frac{\partial[B_s^{TL}]^T}{\partial\{d\}}\{S\}dV \tag{4.24}$$

where $\{S\}$ is the vector of Second Piola-Kirchoff stresses expressed as (Bathe; 2014)

$$\begin{Bmatrix} S_{xx} \\ S_{yy} \\ S_{zz} \\ S_{xy} \\ S_{yz} \\ S_{xz} \end{Bmatrix} = \Big[D\Big] \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{xy} \\ \varepsilon_{yz} \\ \varepsilon_{xz} \end{Bmatrix} \tag{4.25}$$

Note that only the non-linear components of the $[B_s^{TL}]$ matrix are dependent on the displacements. Thus (4.24) can be rewritten as

$$\Big[K_{sNL}^{TL}\Big] = \int_V \frac{\partial[B_{sNL}^{TL}]^T}{\partial\{d\}}\{S\}dV \tag{4.26}$$

Knowing that $[B_{sNL}^{TL}]$ can be expressed by $[A][G]$, allows (4.26) to be rewritten as follows

$$\begin{aligned}\left[K_{sNL}^{TL}\right] &= \int_V [G]^T \frac{\partial [A]^T}{\partial \{d\}} \{S\} dV \\ &= \int_V [G]^T [H] \frac{\partial \{\theta\}}{\partial \{u\}} dV \\ &= \int_V [G]^T [H][G] dV\end{aligned} \tag{4.27}$$

where the matrix $[H]$ is given by

$$\left[H\right] = \begin{bmatrix} S_{xx}[I] & S_{xy}[I] & S_{xz}[I] \\ S_{yx}[I] & S_{yy}[I] & S_{yz}[I] \\ S_{zx}[I] & S_{yx}[I] & S_{zz}[I] \end{bmatrix} \tag{4.28}$$

and $[I]$ is the third order identity matrix. Note that since the stress matrix is symmetric, the $[H]$ matrix will also be symmetric. The complete tangent stiffness matrix can now be expressed numerically as

$$\left[K_s^{TL}\right] = \sum_{i=1}^{nGP} \left[[B_s^{TL}]^T [D][B_s^{TL}] + [G]^T [H][G]\right] det\left[J\right] \omega_i \tag{4.29}$$

where, in the case of isotropic elasticity (Bathe; 2014),

$$\left[D\right] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 & 0 & 0 & 0 \\ \nu & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & k\frac{1-\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & k\frac{1-\nu}{2} \end{bmatrix} \tag{4.30}$$

k is the shear correction factor in (4.30). Note that the zero entries appearing in row and column 3 of $[D]$ above account for the assumption of zero stresses acting

normal to the mid-section of the shell element.

4.2 Shell element layering

In this section, the previously described shell element is extended to incorporate the potential for multiple layers through the thickness. This gives the opportunity to use different material properties across the shell to replicate the heterogeneous (and potentially anisotropic) behaviour of the myocardium.

To achieve this, integration along the thickness direction is undertaken using the standard *Gaussian scheme*. However, the process is repeated as many times as there are layers in the shell element (so for a 2 layer shell element, all 8 *integration points* from the $2 \times 2 \times 2$ *Gaussian scheme* would be computed twice). ζ differs for each computation at each *integration point*. The ζ value varies depending on the number of layers and the thickness of each layer in the shell element (Jensen et al.; 2001)

$$\zeta = -1 + \frac{1}{h_t}(2h_i - h_l(1 - \zeta_l)) \tag{4.31}$$

where h_t is the total thickness of the element, h_i is the sum of the thicknesses of the layers up to the current layer being computed, h_l is the thickness of the current layer and ζ_l is the value of ζ for a single layer Gauss point. Equation (4.31) appears in Line 35 in Listing 4.2 where the local ζ is modified for each Gauss point loop.

In addition to ζ being modified, the stiffness matrix $[\bar{K_s^{TL}}]$ and internal force vector $\{\bar{f}_{int}\}$ have to be adjusted as follows

$$\left[\bar{K_s^{TL}}\right] = \frac{h_l}{h_t}\left[K_s^{TL}\right] \tag{4.32}$$

$$\left\{\bar{f}_{int}\right\} = \frac{h_l}{h_t}\left\{f_{int}\right\} \tag{4.33}$$

4.3 Procedure for a 3D TL Shell FE analysis

```
[coord,etpl,fext0,bc,ngp,lstps,NRitmax,NRtol,ndim,D,dxr]=C_EL;
[nels,nen]=size(etpl); nodes=size(coord,1);
nDoF=nodes*ndim; neDoF=(nen*ndim)^2; lay=size(coord,2)-3;
krow=zeros(neDoF*nels,1); kcol=krow; kval=krow;
uvw=zeros(nDoF,1); uvwold=uvw; fint=uvw; react=uvw;
fd=(1:nDoF); fd(bc(:,1))=[];
epsEn=zeros(6,ngp,lay,nels); epsE=epsEn; sigN=epsEn; sig=epsEn;
oL=zeros(ngp,9,lay,nels); L=oL;
for lstp=0:lstps
  fext=(lstp/lstps)*fext0; oobf=react+fext-fint;
  oobfnorm=2*NRtol; NRit=0;
  while ((NRit<NRitmax)&&(oobfnorm>NRtol))
    NRit=NRit+1; fint=zeros(nDoF,1); dreact=fint; dduvw=fint;
    if lstp>=1
      Kt=sparse(krow,kcol,kval,nDoF,nDoF);
      dduvw(bc(:,1))=(1+sign(1-NRit))*bc(:,2)/lstps;
      dduvw(fd)=Kt(fd,fd)\(oobf(fd)-Kt(fd,bc(:,1))*dduvw(bc(:,1)));
      dreact(bc(:,1))=Kt(bc(:,1),:)*dduvw-oobf(bc(:,1));
    end
    uvw=uvw+dduvw; react=react+dreact; duvw=uvw-uvwold;
    for nel=1:nels
      ed=reshape(ones(ndim,1)*etpl(nel,:)*ndim-...
                 (ndim-1:-1:0).'*ones(1,nen),1,nen*ndim);
      [ke,felem,epsE(:,:,:,nel),sig(:,:,:,nel),L(:,:,:,nel)]=...
      Shell_TL(coord(etpl(nel,:),:),uvw(ed),duvw(ed),ngp,...
        epsEn(:,:,:,nel),D(:,:,:,:,nel),dxr(:,:,:,:,nel),...
        sigN(:,:,:,nel),oL(:,:,:,nel));
      if lstp==0; krow((nel-1)*neDoF+1:nel*neDoF)=...
                    reshape(ed.'*ones(1,nen*ndim),neDoF,1);
                  kcol((nel-1)*neDoF+1:nel*neDoF)=...
                    reshape(ones(nen*ndim,1)*ed  ,neDoF,1);
      end
      kval((nel-1)*neDoF+1:nel*neDoF)=reshape(ke,neDoF,1);
      fint(ed)=fint(ed)+felem;
    end
    oobf=fext+react-fint; oobfnorm=norm(oobf)/norm(fext+react+eps);
    fprintf('%4i %4i %6.3e\n',lstp,NRit,oobfnorm);
  end
  uvwold=uvw; epsEn=epsE; sigN=sig; oL=oL+L;
end
```

Listing 4.1: 40 line MATLAB script expressing the 3D *Total Lagrangian* Shell FE analysis.

```
function [ke,fint,epsE,sig,L]=...
         Shell_TL(nodeData,uvw,duvw,ngp,epsEn,D,dxr,sigN,oL)
nen=size(nodeData,1); lay=size(nodeData,2)-3;
coord=nodeData(:,1:3); t=nodeData(:,4:4+lay-1); T=sum(t,2);
epsE=zeros(6,ngp,lay); sig=epsE; L=zeros(ngp,9,lay);
[wp,GpLoc]=GpPos(ngp); Vn=zeros(nen,3);
ke=zeros(nen*5); fint=zeros(nen*5,1);
ey=[0 1 0].'; ez=[0 0 1].';
V1=zeros(nen,3); V2=V1; g1=V1; g2=V1;
xsi=[-1; -1; -1; 0; 1; 1; 1; 0; 0];
eta=[-1; 0; 1; 1; 1; 0; -1; -1; 0];
dNr=dershapefunc2D(xsi,eta);
dnr=dNr(1:2:end,:); dns=dNr(2:2:end,:);
disp=reshape(uvw-duvw,5,[])'; elcoord=coord+disp(:,1:3);
for n=1:nen
  Vn(n,:)=cross((dnr(n,:)*elcoord)/norm(dnr(n,:)*elcoord),...
                (dns(n,:)*elcoord)/norm(dns(n,:)*elcoord));
end
for n=1:lay
  for node=1:nen
    if abs(ey)-abs(Vn(node,:)')<1e-6
      V=cross(ez,Vn(node,:).').';
    else
      V=cross(ey,Vn(node,:).').';
    end
    V1(node,:)=V/norm(V);
    V2(node,:)=cross(Vn(node,:).',V1(node,:).').';
    g1(node,:)=-0.5*T(n)*V2(node,:);
    g2(node,:)= 0.5*T(n)*V1(node,:);
  end
  for gp=1:ngp
    xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zeta=GpLoc(gp,3);
    N  =shapefunc(xsi,eta);
    tgp=N*t(:,n); Tgp=N*T;
    zet=-1+(2*sum(N*t(:,1:n))-N*(t(:,n).*(1-zeta)))/(N*T);
    dNr=dershapefunc2D(xsi,eta);
    detJ=det(dxr(:,:,gp,n));
    [B, GNL, epsEt,L(gp,:,n)]=...
      formBmatrix(N,dNr,zet,coord,dxr(:,:,gp,n),...
                  g1,g2,duvw,oL(gp,:,n));
    epsE(:,gp,n)=epsEt+epsEn(:,gp,n);
    sig(:,gp,n)=(D(:,:,gp,n)*epsEt)+sigN(:,gp,n);
    H=[sig(1,gp,n)*eye(3) sig(4,gp,n)*eye(3) sig(6,gp,n)*eye(3);
       sig(4,gp,n)*eye(3) sig(2,gp,n)*eye(3) sig(5,gp,n)*eye(3);
       sig(6,gp,n)*eye(3) sig(5,gp,n)*eye(3) sig(3,gp,n)*eye(3)];
    ke=ke+((B'*D(:,:,gp,n)*B)+(GNL'*H*GNL))*detJ*wp(gp)*(tgp/Tgp);
    fint=fint+B'*sig(:,gp,n)*detJ*wp(gp)*(tgp/Tgp);
  end
end
```

Listing 4.2: Function file required for the *Total Lagrangian* Shell FE analysis.

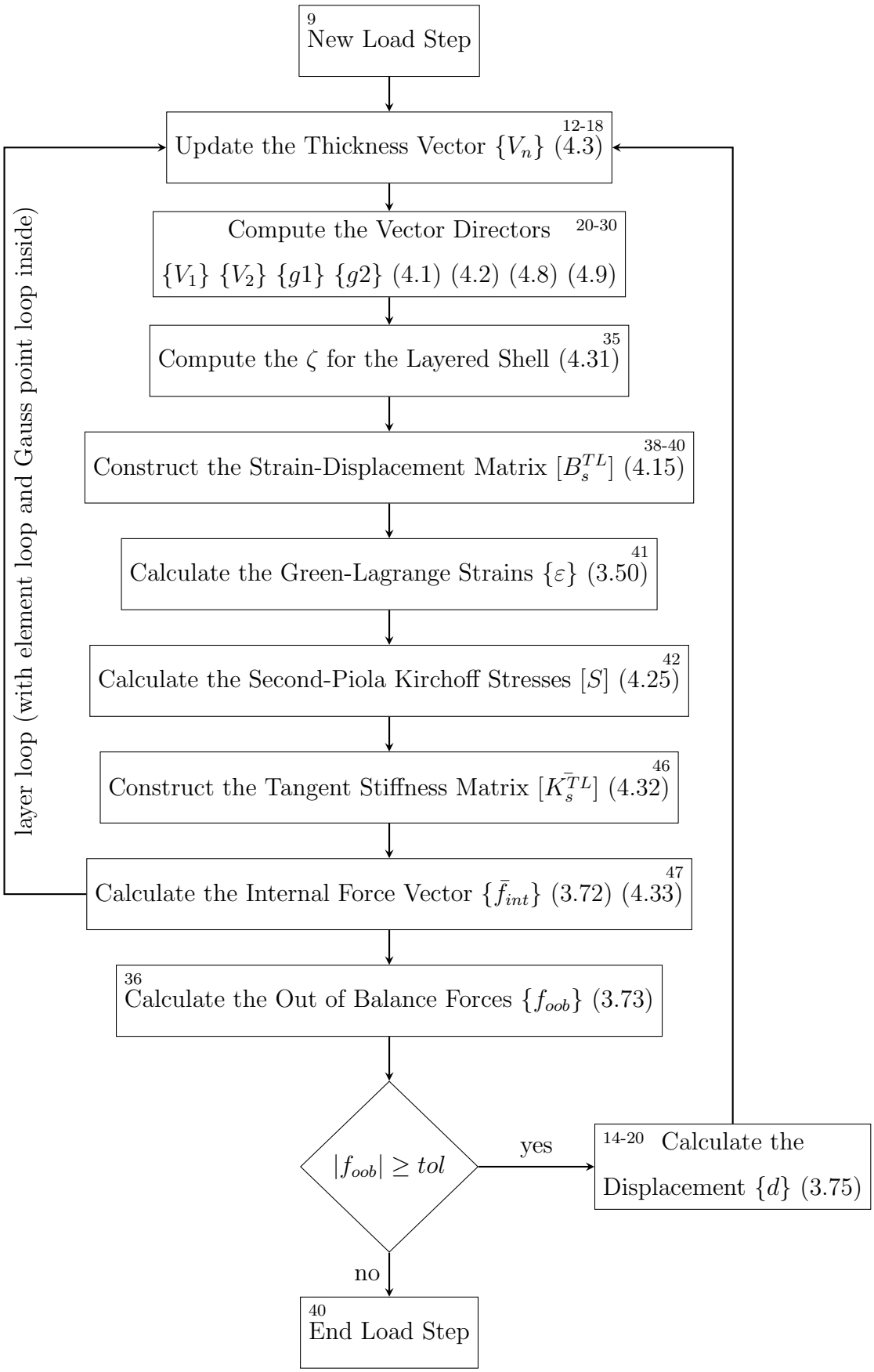


Figure 4.2: Flow diagram depicting the sequential operations within a single load step for the *Total Lagrangian* layered shell FE analysis.

Note that the constitutive model in *Total Lagrangian* approach always refers to the initial configuration, so the $\{V_n\}$ vector used to compute the constitutive matrix should not be updated if the constitutive matrix were formed in the inner loop. In the listings here, the constitutive matrix is formed and stored early in the analysis, before entering time-stepping or *Newton-Raphson* loops. This has the consequences of reducing the computational time but increasing the memory required for the simulation. The data input format for Listing 4.1 is given in the Table below.

Variable	Matrix Sizing	Format
`coord`	$nen \times (3 + nlay)$	$\begin{bmatrix} X_1 & Y_1 & Z_1 & h_1 \\ X_2 & Y_2 & Z_2 & h_2 \\ \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & h_n \end{bmatrix}$ (in m)
`etpl`	$nels \times nen$	$\begin{bmatrix} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 & n_7 & n_8 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$
`fext0`	$(nen \times dof) \times 1$	$\begin{Bmatrix} fex0_1 \\ fext0_2 \\ \vdots \end{Bmatrix}$ (in N)
`bc`	$nbc \times 2$	$\begin{bmatrix} node\_number & node\_displacement \\ \vdots & \vdots \end{bmatrix}$
`ngp`	Scalar	ngp
`lstps`	Scalar	$lstps$
`NRitmax`	Scalar	$NRitmax$
`NRtol`	Scalar	$NRtol$
`ndim`	Scalar	5
`D`	$6 \times 6 \times ngp \times nlay \times nels$	5-D matrix (in Pa)
`dxr`	$3 \times 3 \times ngp \times nlay \times nels$	5-D matrix

Table 4.1: Input format for the layered geometrically non-linear 9-noded shell analysis, with example values for `ngp`, `lstps`, `NRitmax` and `NRtol`.

where nen is the number of nodes defining each element, $nlay$ is the number of layers within each shell element, dof gives the number of degrees of freedom per node, $nels$ is the total number of elements and nbc is the number of degrees of freedom with displacement constraints. The output format is shown in Table 4.2, with u, v and w in metres and ϕ and ψ in radians, with positive values indicating clockwise rotation about $\{V_1\}$ and $\{V_2\}$ (when viewed from the node) respectively.

Variable	Matrix Sizing	Format
`uvw`	$(nen \times dof) \times 1$	$\begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ \phi_1 \\ \psi_1 \\ \vdots \end{Bmatrix}$

Table 4.2: Output from the layered geometrically non-linear 9-noded shell analysis.

4.4 Numerical examples

Two benchmark problems (Sze et al.; 2004) are used to assess the accuracy of the Total Lagrangian large deformation shell finite element analysis code. Both problems consider the deformation of the same very slender cantilever examined in the previous chapter, using hexahedral finite elements. One case involves a transverse end load whereas the other case involves the application of an end moment.

Figure 4.3: Cantilever subjected to an end point load.

Length, L	10 m
Width, b	1 m
Depth, d	0.1 m
Young's modulus, E	1.2 MPa
Poisson's ratio, ν	0
Force applied, F	4 kN

Table 4.3: Input parameters for the non-linear shell cantilever exposed to an end load.

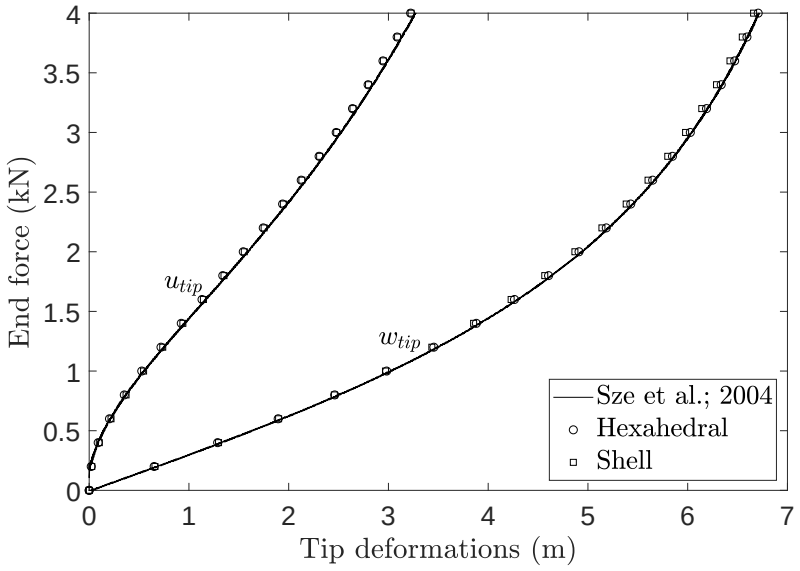


Figure 4.4: Load deflection curves for the cantilever exposed to an end point load.

The first problem was set-up as shown in Figure 4.3 with its parameters as listed in Table 4.3. This problem was simulated with both hexahedral (see Section 3.2.4) and shell elements. 500 hexahedral elements were used, giving a computational run time of 305s while only 8 shell elements (having a computational run time of 28s) were required to achieve similar results. Both simulations are in close agreement with the benchmark solution given by (Sze et al.; 2004). Further increasing the number of elements would increase the computational run time, but not give significant improvements to the results. The input file for this problem is shown in Listing II.4.

The second problem was set-up as shown in Figure 4.5, with the input parameters as listed in Table 4.4. The corresponding results are shown in Figure 4.6.

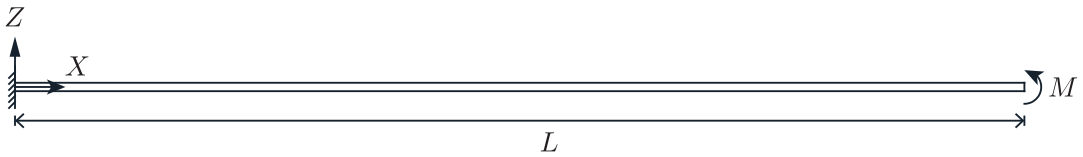


Figure 4.5: Cantilever subjected to an end moment.

Length, L	12 m
Width, b	1 m
Depth, d	0.1 m
Young's modulus, E	1.2 MPa
Poisson's ratio, ν	0
Moment applied, M	$\frac{25\pi}{3}$ kNm

Table 4.4: Input parameters for the non-linear shell cantilever exposed to an end moment.

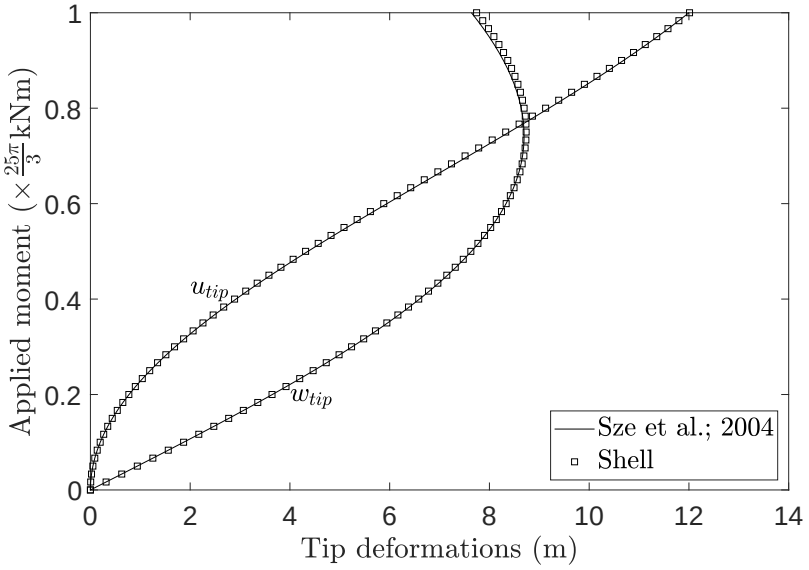


Figure 4.6: Load deflection curves for cantilever exposed to an end moment. Note that all measures on the ordinate should be multiplied by $\frac{25\pi}{3}$ to provide the moment in kNm.

It can be seen from the results obtained, that the shell elements are significantly more effective at capturing the bending behaviour (requiring far fewer elements for an equivalent accuracy). Henceforth, this thesis makes use of shell elements (Total Lagrangian formulation) in order to simulate the behaviour of the myocardium during a cardiac cycle in an attempt to reduce the computational effort required in comparison with low-order hexahedral finite elements. This new feature is a significant development, since all existing finite element simulations of the heart have made use of the simpler but less efficient, tetrahedral or hexhahedral elements. The

input file for this problem is shown in Listing II.5.

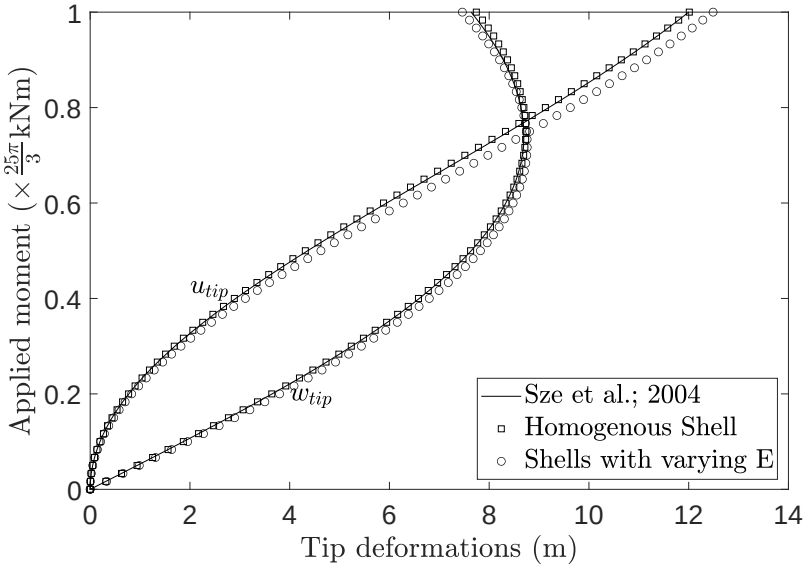


Figure 4.7: Load deflection curves for cantilever with varying Young's modulus across its depth, exposed to an end moment. Note that all measures on the ordinate should be multiplied by $\frac{25\pi}{3}$ to provide the moment in kNm. The solution from Sze *et al.* (Sze et al.; 2004) refers to the homogeneous case.

The second problem (cantilever with an end moment) was run once again with a softer 3-layer representation (with through-thickness elasticities of $0.5E$: E: $2E$ where E indicates a Young's modulus of 1.2 MPa). For this particular loading (where the shell elements are expected to curl inwards), having a softer material on the upper part (inner surface of the curling) of the shell would cause more deformation. The results agree with the expected behaviour giving confident to the code working as intended. It was originally planned in this thesis to make use of a layered approach to account for the differing muscle fibre orientations across the wall of the myocardium. However, upon discovery of the Torrent-Guasp spiral band idealisation (where the band crosses itself in a helical loop), it was recognised that a layered approach would not be required for the analyses intended.

In this chapter, the new FE code was tested against popular benchmark cases which

have demonstrated that the shell formulation requires significantly less computational run time to achieve similar results as a hexahedral FE prediction. In the following chapter, an approach to simulate the myocardium excitation will be described. The electrophysiology will then be combined with the shell mechanical formulation (described in this chapter) to achieve a coupled electro-mechanical capability to model the heart. Although a layered shell formulation has briefly been looked at, that approach will not be used in the later chapters as the Torrent-Guasp model (partially) captures the variation in the fiber orientation across the wall.

References

Ahmad, S. (1969). *Curved Finite Elements in the Analysis of Solid, Shell and Plate Structures*, PhD thesis, University College of Swansea.

Bathe, K.-J. (2014). *Finite Element Procedures*, second edn, Prentice Hall, Pearson Education, Inc.

Bathe, K.-J. and Bolourchi, S. (1980). A geometric and material nonlinear plate and shell element, *Computers & Structures* **11**: 23–48.

Jensen, L. R., Rauhe, J. C. and Stegmann, J. (2001). *Finite Elements for Geometric Non-Linear Analysis of Composite Laminates and Sandwich Structures*, PhD thesis, Aalborg University.

Laulusa, A., Bauchau, O., Choi, J.-Y., Tan, V. and Li, L. (2006). Evaluation of some shear deformable shell elements, *International Journal of Solids and Structures* **43**: 5033–5054.

Parisch, H. (1995). A continuum-based shell theory for non-linear applications, *International Journal for Numerical Methods in Engineering* **38**: 1855–1883.

Sze, K., Liu, X. and Lo, S. (2004). Popular benchmark problems for geometric nonlinear analysis of shells, *Finite Elements in Analysis and Design* **40(11)**: 1551–1569.

5

Modelling the coupled electro-mechanics

There are a great many local, sub-microscopic, electro-chemical processes that occur causing the heart muscles to contract, as described in Chapter 2. To design a meaningfully predictive deformation model of the heart, one would need to express those processes in some way. This chapter explains the approach used here to replicate the transmission of the action potential through the myocardium, leading to the muscle contraction in a phased manner.

5.1 Electro-chemical model

An *electro-physiological model*, which aims to replicate the processes that occur within the heart, comprises two main parts. The first component is often referred to as the *ionic model*, which is used to replicate the behaviour of the cell's membrane potential when it first receives the stimulus to initiate contraction. The second component is the *electro-propagation model* which describes the trasmission of that membrane potential throughout the heart upon receiving the stimulus.

5.1.1 Ionic models

The first attempt at simulating nerve cell behaviour was probably undertaken by Hogkin-Huxley who in 1952 studied a squid axon (Hodgkin and Huxley; 1952). There have been many improvements to this model since its inception and a number of alternative ways to simulate nerve cell behaviour, such as Noble (Noble; 1962), Beeler and Reuter (Beeler and Reuter; 1977) and Luo and Rudy (Luo and Rudy; 1991).

These models take into account the intracellular movement of ion concentrations so as to be able to calculate the membrane potential of cardiac cells at any stage (Beeler and Reuter; 1977) (Luo and Rudy; 1991). Examples of this type of model include works by Noble (Noble; 1962), Beeler and Reuter (Beeler and Reuter; 1977) and Luo and Rudy (Luo and Rudy; 1991). Such models could provide a good prediction of the instance at which cardiac muscle tissue starts contracting. With more advanced models, the changes in the ion concentrations at the cellular level could be calculated. However, these models are computationally expensive, because of the many factors which need to be incorporated over a number of length scales. An average adult human heart will consist of around 10^9 cells and 10^{26} atoms (National Geographic; 2013). By way of comparison, the advanced heart simulation carried out by the Japanese's *Supercomputational Life Science* team made use of approximately 10^6 finite elements (thus modelling the heart on a macroscopic, not atomic or cellular level). Such an analysis already required 17 hours of run-time on a powerful high performance computer (Fujitsu; 2015). It is evidently currently not possible to simulate the whole heart at an atomistic or cellular level even if one had sufficiently detailed information describing the organ at those scales.

5.1.2 Phenomenological models

To simplify the processes occurring at a cellular level without dramatically compromising the accuracy, *phenomenological macroscopic models* have been developed. Examples of these include the FitzHugh-Nagumo model (Fitzhugh; 1961) (Nagumo et al.; 1962) and Aliev and Panfilov model (Aliev and Panfilov; 1996) (Nash and

Panfilov; 2004). These attempt to duplicate the membrane potential signal observed in the muscle tissue with the assumption that all nerves essentially behave the same way in a given region. By removing the need to simulate the ion exchange, these models are significantly more efficient (Fitzhugh; 1961) (Nagumo et al.; 1962). This enables the heart simulation to be modelled at the *mm* (or *cm*) level, rather than at the micron or atomistic levels.

The model that this thesis uses to simulate the ventricular muscles' membrane potential follows the Aliev-Panfilov formulation (Aliev and Panfilov; 1996). This is designed to capture the action potential behaviour in the ventricular tissue with just 2 parameters: the normalized membrane potential, Φ, and a recovery variable r. The latter is used to adjust the *repolarization* phase (to reduce the normalized membrane potential, Φ, back to its initial condition) of an action potential. Both of these variables are dimensionless in the model. The true action potential can be recovered at any stage using the transformation $V^{AP}= (100\Phi - 80)mV$ (Göktepe and Kuhl; 2009).

In the Aliev-Panfilov model, the membrane potential is driven by a source term, f^{Φ} which is expressed as follows (Göktepe and Kuhl; 2009)

$$f^{\Phi} = c\Phi \Big(\Phi - \alpha\Big) \Big(1 - \Phi\Big) - r\Phi + I_{ex} \tag{5.1}$$

where I_{ex} is the external excitation introduced, c is the magnitude that controls the membrane strength and α being the threshold for the excitation. To replicate the repolarization phrase of the cardiac tissue, the membrane potential needs to be brought back to its resting state after a period of time. This is achieved by the recovery variable function, f^{r}, described as follows

$$f^{r} = \Big(\gamma + \tfrac{\mu_1 r}{\mu_2 + \Phi}\Big) \Big(-r - c\Phi \Big(\phi - B - 1\Big)\Big) \tag{5.2}$$

where γ, B, μ_1 and μ_2 are constants that further adjust the duration of the excitation. Higher values of γ, B and μ_1 reduce the excitation duration whereas higher values in μ_2 prolong the excitation duration. Due to the non-linearity of (5.2), a

Newton-Raphson iteration is necessary to solve for the recovery variable r at each time step. The residual error within an iteration, R^r, is given as follows

$$\begin{aligned} R^r &= {}^{t+\Delta t}r - {}^{t}r - f^r \Delta t \\ &= {}^{t+\Delta t}r - {}^{t}r - \left(\gamma + \tfrac{\mu_1 r}{\mu_2 + \Phi}\right)\left(- {}^{t+\Delta t}r - c\Phi\left(\Phi - B - 1\right)\right)\Delta t \end{aligned} \tag{5.3}$$

where Δt is the time-step size. Note that although the analysis involves a time dependency, any inertial effects are ignored in all that follows. This is the normal approach adopted by researchers in this field, although Costabal *et al.* (Costabal et al.; 2017) recently showed that the mass times acceleration term does have some influence and therefore could be included in the mechanical analysis. Expression (5.3) can be linearized by applying an implicit backward-Euler time-stepping scheme and finite difference interpolation to arrive at (Göktepe and Kuhl; 2009)

$$\frac{\partial R^r}{\partial r} = 1 + \left(\gamma + \tfrac{\mu_1 \, {}^{t+\Delta t}r}{\mu_2 + \Phi}\left(- {}^{t+\Delta t}r - c\Phi\left(\Phi - B - 1\right)\right)\right)\Delta t \tag{5.4}$$

The recovery variable is then updated as follows

$${}^{t+\Delta t_{i+1}}r = {}^{t+\Delta t_i}r - \left(\frac{\partial R^r}{\partial r}\right)^{-1} R^r \tag{5.5}$$

where the i in ${}^{t_{i+1}}r$ refers to the i^{th} Newton-Raphson iteration. The membrane potential source term needs to also be solved iteratively, with the residual of the membrane potential source term, R^Φ, given as follows

$$R^\Phi = \frac{{}^{t+\Delta t}\Phi - {}^{t}\Phi}{\Delta t} - f^\Phi \tag{5.6}$$

and its linerization

$$\frac{\partial R^\Phi}{\partial \Phi} = \frac{1}{\Delta t} - \frac{\partial f^\Phi}{\partial \Phi} \tag{5.7}$$

where

$$\frac{\partial f^{\Phi}}{\partial \Phi} = c\left(-3\Phi^2 + 2\Phi\left(1-\alpha\right)+\alpha\right) - r - \Phi\frac{\partial r}{\partial \Phi} \tag{5.8}$$

The last term in (5.8)

$$\frac{\partial r}{\partial \Phi} = -\left(\frac{\partial R^r}{\partial \Phi}\right)^{-1}\frac{\partial R^r}{\partial \Phi} \tag{5.9}$$

where

$$\frac{\partial R^r}{\partial \Phi} = -\left(\left(\gamma + \frac{\mu_1\,{}^{t+\Delta t}r}{\mu_2+\Phi}\right)c\left(2\Phi - B - 1\right) - \frac{\mu_1\,{}^{t+\Delta t}r}{(\mu_2+\Phi)^2}\left({}^{t+\Delta t}r + c\Phi\left(\Phi - B - 1\right)\right)\right)\Delta t \tag{5.10}$$

Finally, the membrane potential can be updated as follows

$${}^{t+\Delta t_{i+1}}\Phi = {}^{t+\Delta t_i}\Phi - \left(\frac{\partial R^{\Phi}}{\partial \Phi}\right)^{-1} R^{\Phi} \tag{5.11}$$

The procedure to update the membrane potential is illustrated in Figure 5.1. Note that the membrane potential is computed at the nodes and then interpolated to the Gauss points for propagation (Niederer et al.; 2011).

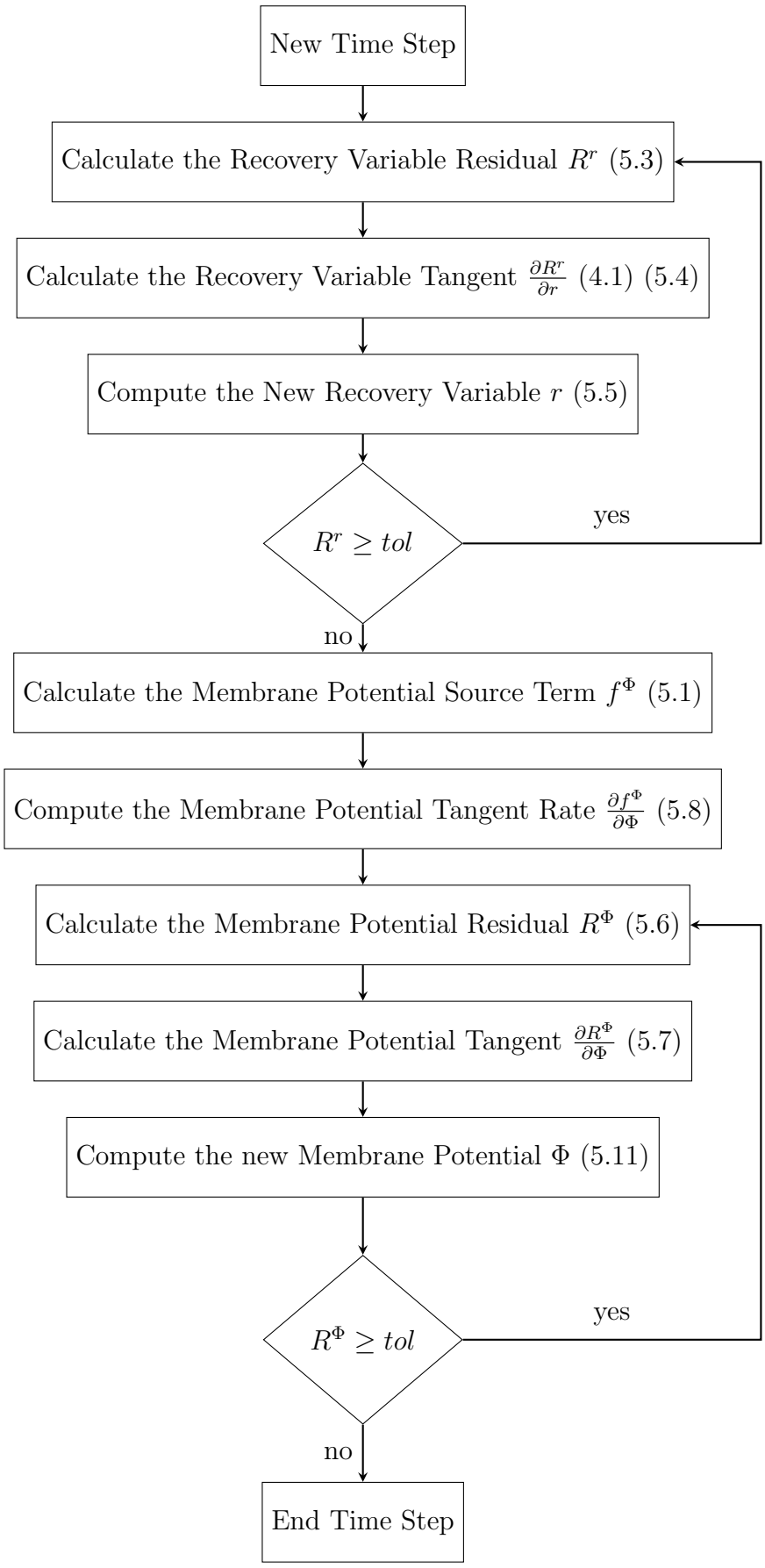


Figure 5.1: Flow diagram depicting the procedure to update the membrane potential at a given location using the Aliev-Panfilov phemonological ionic model.

5.2 Electro-propagation models

There are two electro-propagation approaches which are widely used: the *Bidomain* model or the *Monodomain* model.

5.2.1 Bidomain model

The *Bidomain* model takes into account the ionic exchanges on a macroscopic scale by using two partial differential equations to simulate both the intra and extra-cellular membrane potentials (Ruth; 2014). These partial differential equations are coupled together with the ordinary differential equations that are used to describe the movement of ions through the cell membrane (Shuaiby et al.; 2011). An example of the two-dimensional bidomain equations to be solved numerically is given below (Roth; 1991).

$$C_m \frac{\partial V_m}{\partial t} = -J_{ion} - \frac{1}{\beta}(D_{ex}\frac{\partial^2 V_e}{\partial x^2} + D_{ey}\frac{\partial^2 V_e}{\partial y^2}) \tag{5.12}$$

$$(D_{ix} + D_{ex})\frac{\partial^2 V_e}{\partial x^2} + (D_{iy} + D_{ey})\frac{\partial^2 V_e}{\partial y^2} = -D_{ix}\frac{\partial^2 V_m}{\partial x^2} - D_{iy}\frac{\partial^2 V_m}{\partial y^2} \tag{5.13}$$

where C_m is the membrane capacitance, V_m is the transmembrane potential, J_{ion} is a non-linear term for the transmembrane potential, β is the ratio of the tissue's surface area to its volume, V_e is the extracellular potential, D_{ix} is the intracellular conductivity parallel to the muscle fibres, D_{ex} is the extracellular conductivity parallel to the muscle fibres, D_{iy} is the intracellular conductivity perpendicular to the muscle fibres and D_{ey} is the extracellular conductivity perpendicular to the muscle fibres.

The intracellular potential, V_i can be determined using

$$V_m = V_i - V_e \tag{5.14}$$

5.2.2 Monodomain model

The *Monodomain* model is a simplified representation of the *Bidomain* model that combines the extra- and intracellular membrane potentials into a singular variable by assuming the conductivities of both spaces are proportional, as shown in (5.15) (Shuaiby et al.; 2011).

$$\frac{\partial V_m}{\partial t} + \frac{1}{C_m \beta}(I_{ion} - I_{app}) = \frac{1}{C_m \beta}(D_\xi \frac{\partial^2 V_m}{\partial x^2} + D_\eta \frac{\partial^2 V_m}{\partial y^2}) \tag{5.15}$$

where D_ξ gives the conductivity along the (local ξ in the case of a shell finite element) muscle fibres direction, D_η is the conductivity perpendicular to the (local η) muscle fibres, I_{ion} is the ionic transmembrane current that is generated from the ionic models and I_{app} is the ionic transmembrane current that is applied externally. It is noticed that the above monodomain differential equation is similar to that used in heat diffusion problems, as the membrane potential propagation depends on the conductivity and membrane capacitance only. The Crank-Nicolson approach is known to be stable for diffusion problems (Thomas; 1995). Hence, in this work, the normalized membrane potential (that is Φ rather than V_m) will be propagated using the Crank-Nicolson time-stepping approach adopted by Lewis (Lewis; 1996).

$$\left\{{}^{t+\Delta t}\Phi\right\} = \left[R_p\right]^{-1} \left[S_p\right] \left\{{}^{t}\Phi\right\} \tag{5.16}$$

where

$$\left[R_p\right] = \left[M_p\right] + \frac{\Delta t}{2} \left[K_p\right] \tag{5.17}$$

$$\left[S_p\right] = \left[M_p\right] - \frac{\Delta t}{2} \left[K_p\right] \tag{5.18}$$

with the *mass* matrix, $[M_p]$ expressed as

$$\Big[M_p\Big] = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\Big[N\Big]^T\Big[N\Big]\,\omega_i\omega_j\omega_k det\,\Big[J\Big] \tag{5.19}$$

where $[N]$ is the shape function matrix and ω are the weights in the *Gaussian rule*, i, j and k refer to the *Gaussian points* in each local direction. The corresponding *stiffness* matrix (this is not a mechanical stiffness but expresses the spatial gradient of the diffusivity), $[K_p]$ is given as

$$\Big[K_p\Big] = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\Big[\tfrac{\partial N}{\partial X}\Big]^T\Big[D_{psh}\Big]\Big[\tfrac{\partial N}{\partial X}\Big]\,\omega_i\omega_j\omega_k det\,\Big[J\Big] \tag{5.20}$$

with $\frac{\partial N}{\partial X}$ given by (4.11) and $[D_{psh}]$ defining the diffusivity matrix rotated to align with the local axes of the shell configuration (ξ, η, ζ), which is given as follows

$$\Big[D_{psh}\Big] = \Big[Q_{sh}\Big]^T\Big[D_p\Big]\Big[Q_{sh}\Big] \tag{5.21}$$

where $[D_p]$ is the diffusivity matrix aligned with respect to the global axes described as follows

$$\Big[D_p\Big] = \begin{bmatrix} D_{XX} & D_{XY} & D_{XZ} & 0 & 0 & 0 \\ D_{YX} & D_{YY} & D_{YZ} & 0 & 0 & 0 \\ D_{ZX} & D_{ZY} & D_{ZZ} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.22}$$

and $[Q_{sh}]$ is the transformation matrix that updates the propagation model to the local shell orientation expressed as follows

$$\left[Q_{sh}\right] = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1m_1 & m_1n_1 & n_1l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2m_2 & m_2n_2 & n_2l_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3m_3 & m_3n_3 & n_3l_3 \\ 2l_1l_2 & 2m_1m_2 & 2n_1n_2 & l_1m_2+l_2m_1 & m_1n_2+m_2n_1 & n_1l_2+n_2l_1 \\ 2l_2l_3 & 2m_2m_3 & 2n_2n_3 & l_2m_3+l_3m_2 & m_2n_3+m_3n_2 & n_2l_3+n_3l_2 \\ 2l_3l_1 & 2m_3m_1 & 2n_3n_1 & l_3m_1+l_1m_3 & m_3n_1+m_1n_3 & n_3l_1+n_1l_3 \end{bmatrix} \tag{5.23}$$

where

$$\begin{aligned} l_1 &= \cos(\{e_x\},\{e_\xi\}); \quad m_1 = \cos(\{e_y\},\{e_\xi\}); \quad n_1 = \cos(\{e_z\},\{e_\xi\}) \\ l_2 &= \cos(\{e_x\},\{e_\eta\}); \quad m_1 = \cos(\{e_y\},\{e_\eta\}); \quad n_1 = \cos(\{e_z\},\{e_\eta\}) \\ l_3 &= \cos(\{e_x\},\{e_\zeta\}); \quad m_1 = \cos(\{e_y\},\{e_\zeta\}); \quad n_1 = \cos(\{e_z\},\{e_\zeta\}) \end{aligned} \tag{5.24}$$

$\{e_\xi\}$, $\{e_\eta\}$ and $\{e_\zeta\}$ are the direction vectors (expressed in global coordinates) of the local element coordinates ξ, η and ζ respectively. $\cos(\{e_x\},\{e_\xi\})$, for example, refers to the cosine of the angle between the x and ξ axes.

5.3 Electro-mechanical coupling

The action potential generated in the previous section need to be linked to the mechanical model in order to induce the deformation. This deformation can be achieved either by (i) introducing stresses that are used to compute strains by means of the material's constitutive model or (ii) introducing strains directly that link to the deformation. These methods are known as *active stress* and *active strain* approaches respectively. It is noted that the two techniques are not equivalent (Ambrosi et al.; 2011).

5.3.1 Active stress

An *active stress* approach introduces stresses based on the current excitation level. This method has been used by a number of researchers in the cardiac simulation field (Nash and Panfilov; 2004) (Niederer et al.; 2006) (Costabal et al.; 2017). The deformations induced by these stresses are dependent on the material model prescribed. This allows some flexibility in designing the deformation resulting from the excitation, but careful consideration of the model parameters is required to ensure numerical stability. Furthermore, those material parameters also have to be tuned in order to provide realistic deformations (Ambrosi et al.; 2011) (Ambrosi and Pezzuto; 2011). One approach to introduce active stresses is given below (Nash and Panfilov; 2004)

$$\Delta T_a = \epsilon\left(V\right)\left(k_{Ta}\Phi - T_a\right) \tag{5.25}$$

where T_a is a scalar active stress (which could be applied to the fiber in a particular orientation), k_{Ta} is the amplitude of the stress and

$$\epsilon\left(V\right) = \begin{cases} \epsilon_0, & \text{if } \Phi \leq 0.05 \\ 10\epsilon_0, & \text{otherwise} \end{cases} \tag{5.26}$$

where $\epsilon_0 = 1$.

5.3.2 Active strain

As an alternative to the *active stress* approach, *active strains* can be introduced directly via the membrane potential. This approach avoids the need to tune the constitutive models because the kinematics of the structure are encoded in the method (Ambrosi and Pezzuto; 2011). While this approach removes any influence from the material model, it is still very useful to explore electrophysiology and geometry effects. Grosberg (Grosberg; 2008) imposed a maximum strain of 15% in her simulations. However, contractile myocardium strain up to 20% have been recorded.

Such a strain can be directly introduced locally at a *Gaussian point* level as follows

$$\varepsilon_a = -0.2\Phi \tag{5.27}$$

where ε_a is the active strain (applied in the longitudinal direction of the fibres) and Φ is the excitation level at the integration point. The active strain is converted to equivalent active stresses (assuming that the strain is applied in the local ξ direction) as follows

$$\Big\{S_a\Big\} = \Big[D\Big] \begin{Bmatrix} \varepsilon_a \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{5.28}$$

where $\{S_a\}$ are the active second Piola-Kirchhoff stresses and $[D]$ is the constitutive matrix constructed such that it aligns with the local fibre direction. The active stresses have to be added to the mechanical stresses when determining the stiffness matrix in (4.29) and the internal forces in (3.72). This is done by replacing the stresses, $\{S\}$ in (4.28) and (3.72) with $\{S_T\}$, where $\{S_T\} = \{S\} - \{S_a\}$. The active forces, $\{f_a\}$, corresponding to the above stresses are computed as follows

$$\Big\{f_a\Big\} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \Big[B\Big]^T \Big\{S_a\Big\} \omega_i\omega_j w_\omega det\,\Big[J\Big] \tag{5.29}$$

It is crucial to isolate the active forces as they act as *external forces* to the mechanical model, resulting in the new normalised *out-of-balance* forces which has to be computed as follows

$$\Big|\Big\{f_{oob}\Big\}\Big| = \frac{\Big|\Big\{f_{oob}\Big\}\Big|}{\Big|\Big\{f_{ext}\Big\} + \Big\{f_a\Big\} + \Big\{f_{react}\Big\}\Big|} \tag{5.30}$$

5.4 Procedure of an electro-mechanical coupled FE

```
[coord,etpl,fext0,bc,ngp,lstps,NRitmax,NRtol,ndim,D,Dp,dxr,Q]=exc;
[nels,nen]=size(etpl); nodes=size(coord,1); nDoF=nodes*ndim;
neDoF=(nen*ndim)^2; krow=zeros(neDoF*nels,1); kcol=krow; kval=krow;
uvw=zeros(nDoF,1); uvwold=uvw; fint=uvw; react=uvw;
fd=(1:nDoF); fd(bc(:,1))=[]; lay=size(coord,2)-3;
epsEn=zeros(6,ngp,lay,nels); epsE=epsEn; sigN=epsEn; sig=epsEn;
Vn=zeros(nen,3,nels); oL=zeros(ngp,9,lay,nels);
Told=zeros(nodes,1); dt=0.05; Kp=zeros(nodes); Mp=Kp;
Ptol=1e-8; rn=Told; Kr=Told;
c=8; alp=0.05; gamma=0.002; miu1=0.2; miu2=0.3; beta=0.15;
Iex=15; iex=zeros(nodes,1);
for lstp=0:lstps
  if lstp==1
    iex(1:5)=Iex;
  end
  if lstp==5
    iex=zeros(nodes,1);
  end
  fext=(lstp/lstps)*fext0;
  oobf=react+fext-fint;oobfnorm=2*NRtol; NRit=0;
  while ((NRit<NRitmax)&&(oobfnorm>NRtol))
    NRit=NRit+1; fint=zeros(nDoF,1);
    dreact=fint; dduvw=fint; fP=fint;
    if lstp>=1
      Kt=sparse(krow,kcol,kval,nDoF,nDoF);
      dduvw(bc(:,1))=(1+sign(1-NRit))*bc(:,2)/lstps;
      dduvw(fd)=Kt(fd,fd)\(oobf(fd)-Kt(fd,bc(:,1))*dduvw(bc(:,1)));
      dreact(bc(:,1))=Kt(bc(:,1),:)*dduvw-oobf(bc(:,1));
      if NRit==1
        R=Mp+0.5*dt*Kp; S=Mp-0.5*dt*Kp;
        Tnew=R\(S*Told);
        Told=Tnew;
      end
    end
    uvw=uvw+dduvw; react=react+dreact; duvw=uvw-uvwold;
    if NRit==1
      for node=1:nodes
        r=rn(node); Rr=1e9; Pl=Told(node); Pln=Told(node);
        while abs(Rr)>=Ptol
          Rr=r-rn(node)-...
             ((gamma+((miu1*r)/(miu2+Pl)))*...
              (-r-c*Pl*(Pl-beta-1)))*dt;
          Kr(node)=...
            1+(gamma+(miu1/(miu2+Pl))*(2*r+c*Pl*(Pl-beta-1)))*dt;
          r=r-Kr(node)\Rr;
        end
        rn(node)=r;
        dRrdP=((gamma+((miu1*r)/(miu2+Pl)))*c*(2*Pl-beta-1)-...
               ((miu1*r)/(miu2+Pl)^2)*(r+c*Pl*(Pl-beta-1)))*dt;
        drdP=-Kr(node)\dRrdP;
        fV=-c*Pl*(Pl-alp)*(Pl-1)-r*Pl   + iex(node);
        dfV=c*(-3*Pl^2+2*(1-alp)*Pl+alp)-r-Pl*drdP;
        Rp=1e9;
        while abs(Rp)>=Ptol
          Rp=((Pl-Pln)/dt)-fV;
          dRpdp=(1/dt)-dfV;
          Pl=Pl-dRpdp\Rp;
        end
        Told(node)=Pl;
      end
```

```
61      end
62      for nel=1:nels
63        ed=reshape(ones(ndim,1)*etpl(nel,:)*ndim-...
64                   (ndim-1:-1:0).'*ones(1,nen),1,nen*ndim);
65        [ke,felem,fPel,epsE(:,:,:,nel),Vn(:,:,nel),sig(:,:,:,nel),...
66         L(:,:,:,nel),kp,mp] = Shell_TL_IS(coord(etpl(nel,:),:),...
67           uvw(ed),duvw(ed),ngp,epsEn(:,:,:,nel),D(:,:,:,:,nel),...
68           Dp(:,:,:,:,nel),dxr(:,:,:,:,nel),sigN(:,:,:,nel),...
69           oL(:,:,:,nel),Told(etpl(nel,:)),Q(:,:,:,:,nel));
70        if lstp==0; krow((nel-1)*neDoF+1:nel*neDoF)=...
71                      reshape(ed.'*ones(1,nen*ndim),neDoF,1);
72                    kcol((nel-1)*neDoF+1:nel*neDoF)=...
73                      reshape(ones(nen*ndim,1)*ed  ,neDoF,1);
74        end
75        kval((nel-1)*neDoF+1:nel*neDoF)=reshape(ke,neDoF,1);
76        fint(ed)=fint(ed)+felem; fP(ed)=fP(ed)+fPel;
77        Kp(etpl(nel,:),etpl(nel,:))=Kp(etpl(nel,:),etpl(nel,:))+kp;
78        Mp(etpl(nel,:),etpl(nel,:))=Mp(etpl(nel,:),etpl(nel,:))+mp;
79      end
80      oobf=fext+react-fint;
81      oobfnorm=norm(oobf)/norm(fext+fP+react+eps);
82      fprintf('%4i %4i %6.3e\n',lstp,NRit,oobfnorm);
83    end
84    uvwold=uvw; epsEn=epsE; sigN=sig; oL=oL+L;
85  end
```

Listing 5.1: 85 line MATLAB script expressing the 3D Electro-Mechanical Coupled *Total Lagrangian* Shell FE analysis.

```
function [ke,fint,fP,epsE,Vn,sig,L,kp,mp]=...
  Shell_TL_IS(nodeData,uvw,duvw,ngp,epsEn,D,Dp,dxr,sigN,oL,Told,Q)
nen=size(nodeData,1); lay=size(nodeData,2)-3;
coord=nodeData(:,1:3); t=nodeData(:,4:4+lay-1); T=sum(t,2);
epsE=zeros(6,ngp,lay); sig=epsE;
[wp,GpLoc]=GpPos(ngp); L=zeros(ngp,9,lay);
ke=zeros(nen*5); fint=zeros(nen*5,1);
ey=[0 1 0].'; ez=[0 0 1].'; fP=fint; kp=zeros(nen); mp=kp;
V1=zeros(nen,3); V2=V1; g1=V1; g2=V1; Vn=zeros(9,3);
xsi=[-1; -1; -1; 0; 1; 1; 1; 0; 0];
eta=[-1; 0; 1; 1; 1; 0; -1; -1; 0];
dNr=dershapefunc2D(xsi,eta);
dnr=dNr(1:2:end,:); dns=dNr(2:2:end,:);
disp=reshape(uvw-duvw,5,[])'; elcoord=coord+disp(:,1:3);
for n=1:9
  Vn(n,:)=cross((dnr(n,:)*elcoord)/norm(dnr(n,:)*elcoord),...
                (dns(n,:)*elcoord)/norm(dns(n,:)*elcoord));
end
for n=1:lay
  for node=1:nen
    if norm(cross(ey,Vn(node,:)'))<1e-6
      V=cross(ez,Vn(node,:)')';
    else
      V=cross(ey,Vn(node,:)')';
    end
    V1(node,:)=V/norm(V);
    V2(node,:)=cross(Vn(node,:).',V1(node,:).').';
    g1(node,:)=-0.5*T(n)*V2(node,:);
    g2(node,:)= 0.5*T(n)*V1(node,:);
  end
  for gp=1:ngp
    xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zeta=GpLoc(gp,3);
    N  =shapefunc(xsi,eta);
    tgp=N*t(:,n); Tgp=N*T;
    zet=-1+(2*sum(N*t(:,1:n))-N*(t(:,n).*(1-zeta)))/(N*T);
    dNr=dershapefunc2D(xsi,eta);
    dNx=dxr(:,:,gp,n)\[dNr; zeros(1,nen)];
    detJ=det(dxr(:,:,gp,n));
    Tl=N*Told;
    if Tl<0; Tl=0; elseif Tl>1; Tl=1; end
    epsP=zeros(6); epsP(1)=0.15*Tl;
    epsPsh=Q(:,:,gp,n)'*epsP*Q(:,:,gp,n);
    sigP=D(:,:,gp,n)*(-epsPsh(1:7:end)');
    [B, GNL, epsEt,L(gp,:,n)]=...
      formBmatrix(N,dNr,zet,coord,dxr(:,:,gp,n),...
                  g1,g2,duvw,oL(gp,:,n));
    epsE(:,gp,n)=epsEt+epsEn(:,gp,n);
    sig(:,gp,n)=(D(:,:,gp,n)*epsEt)+sigN(:,gp,n);
    sigT=sig(:,gp,n)+D(:,:,gp,n)*(epsPsh(1:7:end)');
    H=[sigT(1)*eye(3) sigT(4)*eye(3) sigT(6)*eye(3);
       sigT(4)*eye(3) sigT(2)*eye(3) sigT(5)*eye(3);
       sigT(6)*eye(3) sigT(5)*eye(3) sigT(3)*eye(3)];
    ke=ke+((B'*D(:,:,gp,n)*B)+(GNL'*H*GNL))*detJ*wp(gp)*(tgp/Tgp);
    fint=fint+B'*sigT*detJ*wp(gp)*(tgp/Tgp);
    fP=fP+B'*sigP*detJ*wp(gp)*(tgp/Tgp);
    kp=kp+dNx'*Dp(:,:,gp,n)*dNx*wp(gp)*detJ*(tgp/Tgp);
    mp=mp+N'*N*wp(gp)*detJ*(tgp/Tgp);
  end
end
```

Listing 5.2: Function file required for the Electro-Mechanical Coupled *Total Lagrangian* Shell FE analysis.

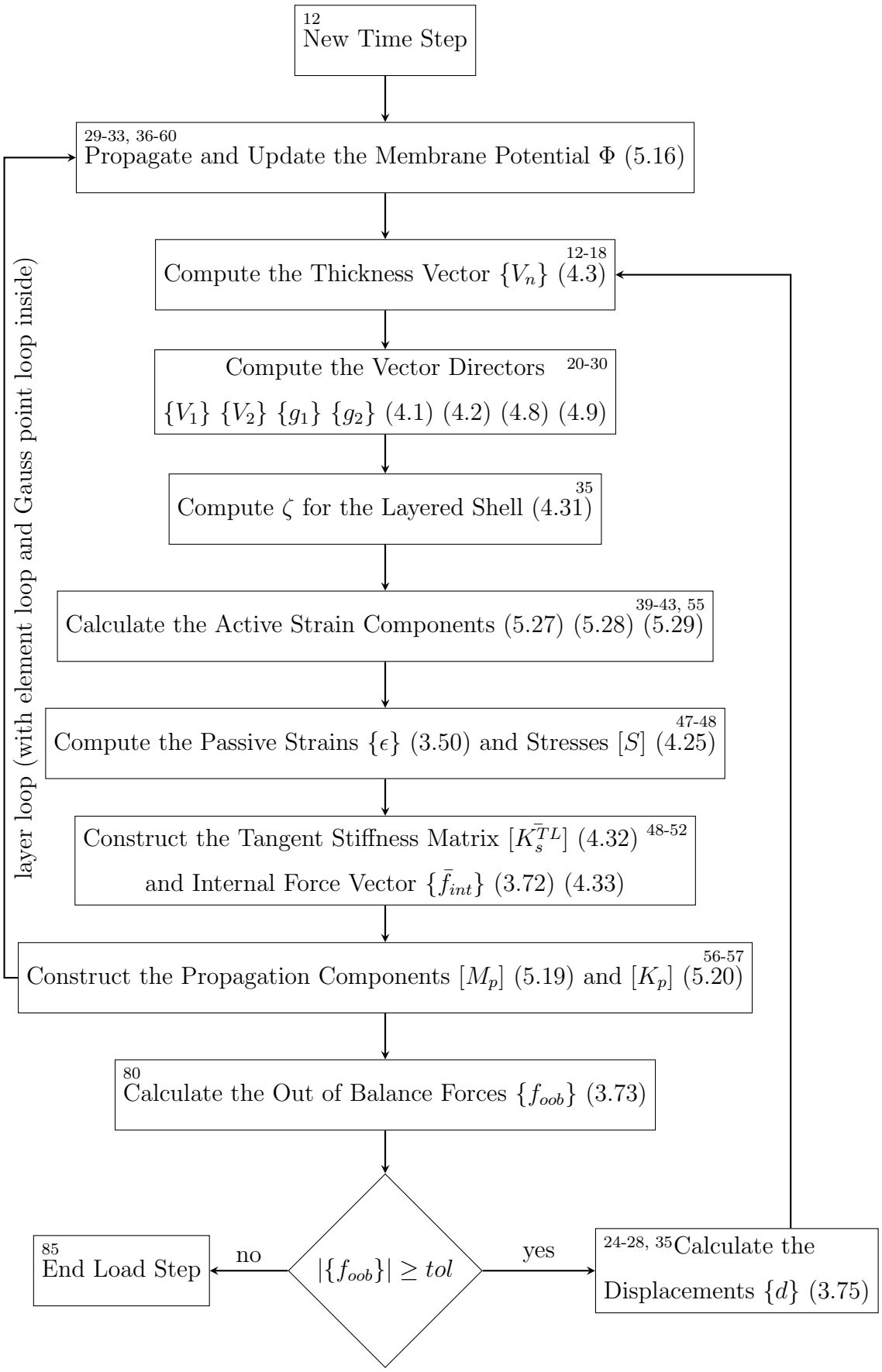


Figure 5.2: Flow diagram depicting the sequential operation within a single time step for the Electro-Mechanical coupled *Total Lagrangian* layered shell FE analysis.

The data input format for Listing 5.1 is given in the Table below.

Variable	Matrix Sizing	Format
`coord`	$nen \times (3 + nlay)$	$\begin{bmatrix} X_1 & Y_1 & Z_1 & h_1 \\ X_2 & Y_2 & Z_2 & h_2 \\ \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & h_n \end{bmatrix}$ (in m)
`etpl`	$nels \times nen$	$\begin{bmatrix} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 & n_7 & n_8 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$
`fext0`	$(nen \times dof) \times 1$	$\begin{Bmatrix} 0 \\ 0 \\ \vdots \end{Bmatrix}$ (in N)
`bc`	$nbc \times 2$	$\begin{bmatrix} node\_number & node\_displacement \\ \vdots & \vdots \end{bmatrix}$
`ngp`	Scalar	ngp
`lstps`	Scalar	$lstps$
`NRitmax`	Scalar	$NRitmax$
`NRtol`	Scalar	$NRtol$
`ndim`	Scalar	5
`D`	$6 \times 6 \times ngp \times nlay \times nels$	5-D matrix (in Pa)
`Dp`	$3 \times 3 \times ngp \times nlay \times nels$	5-D matrix (in $\frac{m^2}{s}$)
`dxr`	$3 \times 3 \times ngp \times nlay \times nels$	5-D matrix
`Q`	$6 \times 6 \times ngp \times nlay \times nels$	5-D matrix

Table 5.1: Input format for the electro-mechanical coupled layered geometrically non-linear 9-noded shell analysis, with example values for `ngp`, `lstps`, `NRitmax` and `NRtol`.

where nen is the number of nodes defining each element, $nlay$ is the number of layers within each shell element, dof gives the number of degrees of freedom per node, $nels$ is the total number of elements, nbc is the number of degrees of freedom with displacement constraints and $[Q]$ is the transformation matrix expressed in (5.23). The output format is shown in Table 4.2, with u, v and w in metres and ϕ and ψ in radians, with positive values indicating clockwise rotation about $\{V_1\}$ and $\{V_2\}$ (when viewed from the node) respectively.

5.5 Numerical examples

The procedure to compute the membrane potential at a local point (see (5.11)) is illustrated in Figure 5.1. The parameters used for the model are listed in Table 5.3,

Variable	Matrix Sizing	Format
`uvw`	$(nen \times dof) \times 1$	$\begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ \phi_1 \\ \psi_1 \\ \vdots \end{Bmatrix}$

Table 5.2: Output from the electro-mechanical coupled layered geometrically non-linear 9-noded shell analysis.

with the code given in Listing 5.3. The stimulus was introduced at the 15^{th} time step (9.6745ms) and removed at the next time step. The corresponding results are shown in Figure 5.3.

c	8
α	0.05
γ	0.002
μ_1	0.2
μ_2	0.3
B	0.15
Time step, Δt	0.645 ms
External stimulus, I_{ex}	30

Table 5.3: Input parameters for the Aliev-Panfilov model for an excitation lasting $400ms$.

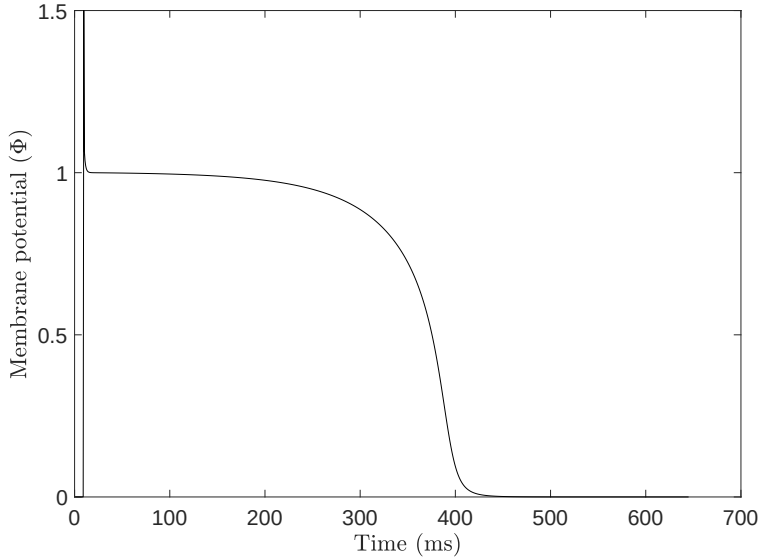


Figure 5.3: Time history of the membrane potential generated by the Aliev-Panfilov model, given the input parameters of Table 5.3.

Once again, note that the normalised membrane potential, Φ, requires the transformation $V^{AP} = (100\Phi - 80)mV$ (Göktepe and Kuhl; 2009) to give the membrane potential in mV. It can be seen from Figure 5.3 that Φ exceeds the value of 1 for a few time steps upon introduction of excitation and then converges to the threshold excitation value of 1, whereafter it starts to decline. It starts rapidly reducing after 350ms. This behaviour is similar to that shown in other literature (Göktepe and Kuhl; 2009) (Göktepe and Kuhl; 2010).

```
c=8; alp=0.05; gamma=0.002; miu1=0.2; miu2=0.3; beta=0.15;
tol=1e-10; dt=0.05;
rn=0; Vn=0; V=0; dVt=0; Ta=0; Iex=30; iex=0;
for ts=1:1000
  if ts==15
    iex=Iex;
  end
  if ts==16
    iex=0;
  end
  r=rn; R=1e9;
  while abs(R)>=tol
    R=r-rn-((gamma+((miu1*r)/(miu2+V)))*(-r-c*V*(V-beta-1)))*dt;
    dRr=1+(gamma+(miu1/(miu2+V))*(2*r+c*V*(V-beta-1)))*dt;
    r=r-dRr\R;
  end
  rn=r;
  dRV=((gamma+((miu1*r)/(miu2+V)))*c*...
       (2*V-beta-1)-((miu1*r)/(miu2+V)^2)*(r+c*V*(V-beta-1)))*dt;
  drV=-inv(dRr)*dRV;
  fV=-c*V*(V-alp)*(V-1)-r*V + iex;
  dfV=c*(-3*V^2+2*(1-alp)*V+alp)-r-V*drV;
  Rp=1e9;
  while abs(Rp)>=tol
    Rp=((V-Vn)/dt)-fV;
    dRpdp=(1/dt)-dfV;
    V=V-dRpdp\Rp;
    1;
  end
  Vn=V;
  plot(ts*dt*12.9,V,'k.'); hold on;
end
```

Listing 5.3: 32 line MATLAB script expressing the membrane potential calculation using the Aliev-Panfilov model.

Two problems were used to verify and validate the electro-propagation model. Both problems consider a $100 \times 10 \times 1mm$ panel, although the analysis is essentially two-dimensional with the 1 mm thickness dimension not featuring in the simulation. In the first benchmark problem, the membrane potential propagates uniformly from left to right across the whole width of the panel. In the second problem, the membrane potential spreads from the lower left corner (initially acting over half of the 10 mm width of the panel). The parameters for these problems are given in Table 5.4 and the input file is shown in Listing 5.4.

Length	100 mm
Width	10 mm
Height	1 mm
Young's modulus, E	1.2 MPa
Poisson's ratio, ν	0
Number of elements across length	10
Number of elements across width	2
Number of elements across height	1
Active strain	15%
Diffusivity	32.25 $\times 10^{-3}$ $m^2 s^{-1}$

Table 5.4: Input parameters for the electro-mechanical coupled analysis of a panel.

```
function [coord,etpl,fext,bc,ngp,lstps,NRitmax,NRtol,...
          ndim,D,Dp,dxr,Q]=exc2
ngp=18; NRitmax=50; NRtol=1e-6; ndim=5; lstps=100;
len=0.1; wid=0.01; thk=0.001; nen=9;
E=1.2e6; nu=0; ka=5/6;
lay=1; E=E*ones(1,lay);
nelx=10; nely=2; nels=nelx*nely;
coord=zeros((nelx*2+1)*(nely*2+1),3+lay);
coord(:,4:4+lay-1)=thk/lay;
etpl=zeros(nels,9); k=1;
for i=1:(nelx*2+1)
  for j=1:(nely*2+1)
    coord(k,1:2)=[(i-1)*len/(2*nelx) (j-1)*wid/(2*nely)]; k=k+1;
  end
end
k=1;
for i=1:nelx
  for j=1:nely
    n1=(i-1)*2*(nely*2+1)+(j-1)*2+1;
    n2=n1+1; n3=n2+1; n4=n3+(nely*2+1); n5=n4+(nely*2+1);
    n6=n5-1; n7=n6-1; n8=n4-2; n9=n8+1;
    etpl(k,:)=[n1 n2 n3 n4 n5 n6 n7 n8 n9];
    oVn(:,:,k)=[zeros(nen,2) ones(nen,1)]; k=k+1;
  end
end
nodes=size(coord,1);
fext=zeros(nodes*ndim,1);
bc=zeros(nodes*5,2); k=0;
for i=1:nodes
  if coord(i,1)==0
    bc(i*5-4,:)=[i*5-4 0]; k=k+1;
    bc(i*5-3,:)=[i*5-3 0]; k=k+1;
    bc(i*5-2,:)=[i*5-2 0]; k=k+1;
    bc(i*5-1,:)=[i*5-1 0]; k=k+1;
    bc(i*5  ,:)=[i*5   0]; k=k+1;
  end
end
bc=sortrows(bc,1);
bc(1:nodes*5-k,:)=[];

D=zeros(6,6,ngp,lay,nels); dxr=zeros(3,3,ngp,lay,nels);
Dp=dxr; Q=D;
ex=[1 0 0].'; ey=[0 1 0].'; ez=[0 0 1].';
Dpn=zeros(6); Dpn(1)=3.225e-4;
for nel=1:nels
  elcoord=coord(etpl(nel,:),1:3);
  t=coord(etpl(nel,:),4:4+lay-1); T=sum(t,2);
  [~,GpLoc]=GpPos(ngp);
```

```
  for n=1:lay
    Dn=(E(n)/(1-nu^2))*[[1 nu; nu 1; 0 0] zeros(3,4);...
      zeros(3,3) eye(3)*ka*(1-nu)*0.5];
    Dn(4,4)=Dn(4,4)/ka;
    for gp=1:ngp
      xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zeta=GpLoc(gp,3);
      N  =shapefunc(xsi,eta);
      zet=-1+(2*sum(N*t(:,1:n))-N*(t(:,n).*(1-zeta)))/(N*T);
      dNr=dershapefunc2D(xsi,eta);
      dxr(1:2,:,gp,n,nel)=...
        (dNr*(elcoord+0.5*zet*(oVn(:,:,nel).*(T*ones(1,3)))));
      dxr(3,:,gp,n,nel)=(0.5*(N.*T')*oVn(:,:,nel));
      VnGp=(N*oVn(:,:,nel)).';
      sdir=dxr(2,:,gp,n,nel)'/norm(dxr(2,:,gp,n,nel)');
      er=cross(sdir,VnGp); er=er/norm(er);
      es=cross(VnGp,er);   es=es/norm(es);
      et=VnGp/norm(VnGp);
      l1=(ex.'*er); m1=(ey.'*er); n1=(ez.'*er);
      l2=(ex.'*es); m2=(ey.'*es); n2=(ez.'*es);
      l3=(ex.'*et); m3=(ey.'*et); n3=(ez.'*et);
      Q(:,:,gp,n,nel)=...
        [      l1*l1       m1*m1       n1*n1...
               l1*m1       m1*n1       n1*l1 ;
               l2*l2       m2*m2       n2*n2...
               l2*m2       m2*n2       n2*l2 ;
               l3*l3       m3*m3       n3*n3...
               l3*m3       m3*n3       n3*l3 ;
             2*l1*l2     2*m1*m2     2*n1*n2...
          l1*m2+l2*m1 m1*n2+m2*n1 n1*l2+n2*l1 ;
             2*l2*l3     2*m2*m3     2*n2*n3...
          l2*m3+l3*m2 m2*n3+m3*n2 n2*l3+n3*l2 ;
             2*l3*l1     2*m3*m1     2*n3*n1...
          l3*m1+l1*m3 m3*n1+m1*n3 n3*l1+n1*l3];
      D(:,:,gp,n,nel)=Q(:,:,gp,n,nel)'*Dn*Q(:,:,gp,n,nel);
      Dpt=Q(:,:,gp,n,nel)'*Dpn*Q(:,:,gp,n,nel);
      Dp(:,:,gp,n,nel)=Dpt(1:3,1:3);
    end
  end
end
```

Listing 5.4: The input file used to feed Listing 5.2 for the 3D Electro-Mechanical Coupled *Total Lagrangian* Shell FE analysis shown in Figure 5.4.

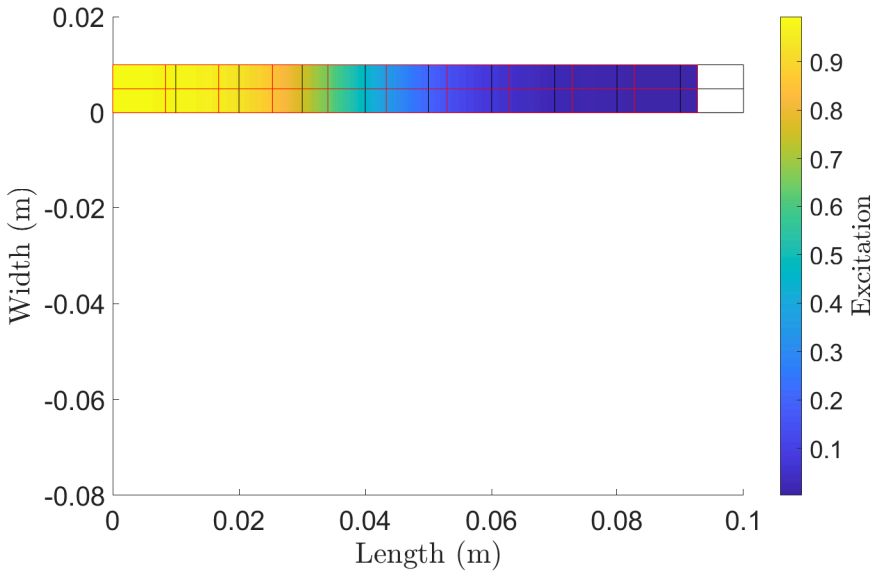


Figure 5.4: Plan view of membrane potential propagation and deformation (at 32.25ms) in a panel experiencing excitation from the left, where yellow regions are fully excited and blue regions are unexcited.

The state of the panel, for the first problem (at 32.25 ms) is shown in Figure 5.4. The image shows a side elevation of the panel, with the yellow region indicating the domain experiencing excitation and the blue region yet to be excited. The panel can be seen to shorten (the black wireframe indicates the initial geometry, whereas and the red lines show the current deformed configuration) following contraction on the left. This successful implementation of excitation causing contraction gives confidence that the codes are working as intended.

The influence of the time-step size and the mesh density were explored for this problem. By only reducing the time-step size, the excitation is proportionally reduced as illustrated in Figure 5.5.

To properly scale the analysis, the excitation magnitude has to be increased by the same amount as Δt is decreased. This gives rise to the solution seen in Figure 5.6. This re-scaling is a consequence of the external excitation (see (5.1)) which needs to balance the rate of change of excitation (which is scaled by the time-step size) in

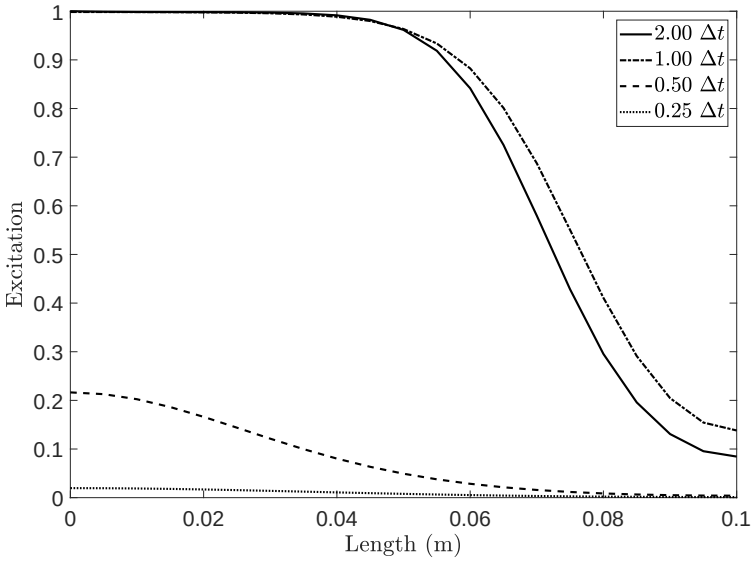


Figure 5.5: Excitation level across the length of the panel (at 32.25ms) following analyses where the time-step size alone is altered. This gives an incorrect set of results.

(5.6).

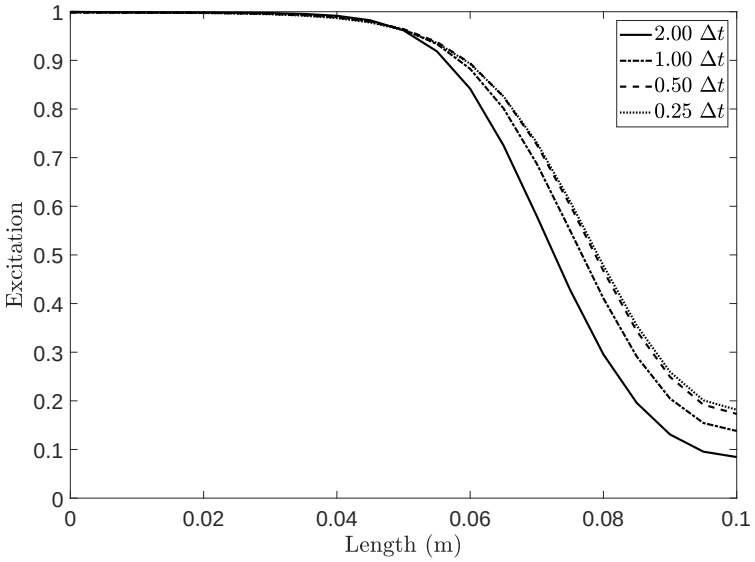


Figure 5.6: Excitation level across the length of the panel (at 32.25ms) for different time-step size with the excitation proportionally scaled.

The following study investigates the effects of changing the mesh density. In order to preserve the physics of the problem, both the time-step size and initial excitation strength need to be proportionally changed when the mesh density is altered. This

is because the strength of the excitation being introduced is dependent on the size of the element as can be seen in (5.16). Figure 5.7 shows the results from the different mesh densities. Doubling the number of elements in the panel from 20 to 40 gives quite similar results, indicating a convergence of the simulations.

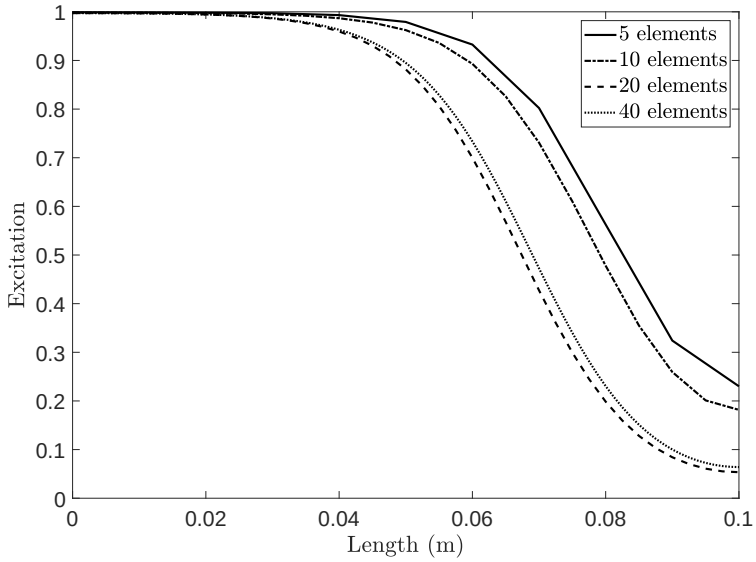


Figure 5.7: Excitation level across the length of the panel (at 32.25ms) for different mesh densities.

In the second problem, where the excitation was introduced across half the width of the left side, a downwards flexure (or rotation) of the panel is seen in Figure 5.8. Note that the diffusivity of the material was defined as being non-zero only in the longitudinal (length) direction. A consequence of this uni-directional diffusion is that the excitation does not propagate in the width direction, leading to the curving down of the panel. The findings from having smaller time-step sizes and greater mesh densities, where the results converged towards a unique solution provides assurement that the FE codes are working as intended. However, it is important to note that the coupling of the electrophysiology model with the mechanical model (both of which are non-linear) introduces additional non-linearities which require some possible fine-tuning in order to undertake the analysis. For example, careful consideration of the initial excitation is required.

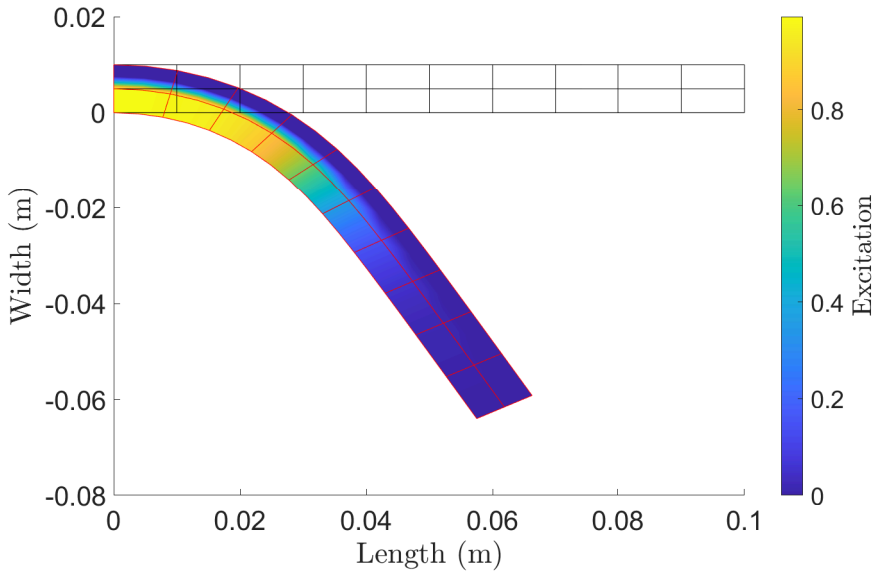


Figure 5.8: Plan view of the membrane potential propagation and deformation (at 32.25ms) in a panel experiencing excitation from the lower left corner, where yellow regions are fully excited and blue regions are unexcited.

In this chapter, the formulation used to model the cardiac excitation is described. This formulation is coupled with the shell formulation from the previous Chapter. Simulations have been carried out to provided confident in the coupled electro-mechanical analysis code by revealing progressive deformations which are consistent with the expectations of the model. In the following chapter, the importance of the Torrent-Guasp model will be explored and the code presented here will be used to model the said model.

References

Aliev, R. R. and Panfilov, A. V. (1996). A simple two-variable model of cardiac excitation, *Chaos, Solitons & Fractals* **7**: 293–301.

Ambrosi, D., Arioli, G., Nobile, F. and Quarteroni, A. (2011). Electromechanical

coupling in cardiac dynamics: The active strain approach, *Society for Industrial and Applied Mathematics* pp. 605–621.

Ambrosi, D. and Pezzuto, S. (2011). Active stress vs. active strain in machanobiology: Constitutive issues, *Journal of Elasticity* **107**: 199–212.

Beeler, G. W. and Reuter, H. (1977). Reconstruction of the action potential of ventricular myocardial fibres, *The Journal Of Physiology* **268(1)**: 177–210.

Costabal, F. S., Concha, F. A., Hurtado, D. E. and Kuhl, E. (2017). The importance of mechano-electrical feedback and inertia in cardiac electromechanics, *Computer methods in applied mechanics and engineering* **320**: 352–368.

Fitzhugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane, *Biophysical Journal* **1(6)**: 445–466.

Fujitsu (2015). Using supercomputers for safer heart surgery, `http://www.fujitsu.com/global/vision/customerstories/university-of-tokyo/`.

Göktepe, S. and Kuhl, E. (2009). Computational modeling of cardiac electrophysiology: A novel finite element approach, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING* **79**: 156–178.

Göktepe, S. and Kuhl, E. (2010). Electromechanics of the heart: A unified approach to the strongly coupled excitationcontraction problem, *Computational Mechanics* **45(2)**: 227–243.

Grosberg, A. (2008). *A Bio-Inspired Computational Model of Cardiac Mechanics: Pathology and Development*, PhD thesis, California Institute of Technology.

Hodgkin, A. and Huxley, A. (1952). A quantitative description of membrane current and it's application to conduction and excitation in nerve, *The Journal of Physiology* **117**: 500–544.

Lewis, R. (1996). *The Finite Element Method in Heat Transfer Analysis*, John Wiley & Sons.

Luo, C. H. and Rudy, Y. (1991). A model of the ventricular cardiac action potential. depolarization, repolarization, and their interaction, *Circulation Research* **68(6)**: 1501–1526.

Nagumo, J., Arimoto, S. and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon, *Proceedings of the IRE* **50(10)**: 2061–2070.

Nash, M. P. and Panfilov, A. V. (2004). Electromechanical model of excitable tissue to study reentrant cardiac arrhythmias, *Progress in Biophysics & Molecular Biology* **85**: 501–522.

National Geographic (2013). How many cells are in your body?, `http://phenomena.nationalgeographic.com/2013/10/23/how-many-cells-are-in-your-body/`.

Niederer, S. A., Kerfoot, E., Benson, A. P., Bernabeu, M. O., Bernus, O., Bradley, C., Cherry, E. M., Clayton, R., Fenton, F. H., Garny, A., Heidenreich, E., Land, S., Maleckar, M., Pathmanathan, P., Plank, G., Rodríguez, J. F., Roy, I., Sachse, F. B., Seemann, G., Skavhaug, O. and Smith, N. P. (2011). Verification of cardiac tissue electrophysiology simulators using an n-version benchmark, *Philosophical Transactions of The Royal Society* **369**: 4331–4351.

Niederer, S., Hunter, P. and Smith, N. (2006). A quantitative analysis of cardiac myocyte relaxation: A simulation study, *Biophysical Journal* **90**: 1697–1722.

Noble, D. (1962). A modification of the hodgkin - huxley equations applicable to purkinje fibre action and pacemaker potentials, *The Journal Of Physiology* **160(2)**: 317–352.

Roth, B. J. (1991). Action potential propagation in a thick strand of cardiac muscle, *Circulation Research* **68(1)**: 162–173.

Ruth, A. (2014). *Electromechanical Large Scale Computational Models of the Ventricular Myocardium*, PhD thesis, Universitat Politècnica de Catalunya.

Shuaiby, S. M., Hassan, M. A. and El-Melegy, M. (2011). Modeling and simulation of the action potential in human cardiac tissues using finite element method, *Journal of Communications and Computer Engineering* **2:3**: 21–27.

Thomas, J. (1995). *Numerical Partial Differential Equations: Finite Difference Methods*, Springer-Verlag.

6

Coupled electromechanical large deformation finite element analyses: simulations of a contracting and relaxing left ventricle

The heart has evolved to become a very efficient pump. Yet a full understanding of the deformation sequence and the consequences of changes in the heart geometry have yet to be fully appreciated. Through use of the coupled electro-mechanical Total Lagrangian large deformation shell finite element code (developed earlier in this thesis), new simulations are offered which help to reveal the complex behaviour. Unlike earlier hexahedral or tetrahedral analyses (on ellipsoidal representations of the left ventricle) by others (Kotikanyadanam et al.; 2010) (Göktepe et al.; 2011), the simulation given here makes use of an idealisation of the helical Torrent-Guasp band model first developed by Grosberg (Grosberg; 2008). The role of the heart geometry (for example, its sphericity) and the influence of stiffer regions (infarct), the ejection fraction, the degreee of twisting and the rate of relaxation are all reported in this Chapter. It is argued that the finite element model offers a useful new computational tool which could help in our understanding of cardiac mechanics and aid the early

diagnosis of cardiac disease.

6.1 Influence of the geometric parameters

It is known that the cardiac muscle contracts up to 20% (Buckberg et al.; 2018) (Sengupta and Narula; 2014) leading to a typical ejection fraction in the left ventricle of between 60 and 70%. Such high ejection fractions are only possible if the muscle bands are arranged in a helical configuration. This finding confirms that the geometrical arrangement of the myocardial band plays an essential role in the pumping efficiency of this organ (Grosberg; 2008).

6.1.1 Helix angle

In this section, the significance of the orientation of the helical muscle fibre band in the left ventricle is explored. The coupled electro-mechanical finite element shell model described in the previous chapter is first arranged into a simple cylindrical form with varying helix band angles (measured from the cylinder long axis) in a series of spirals to study the effect of this angle on the ejection fraction following band shortening. In this study right helical spirals are formed. Thus the helix angles are negatively valued. A left helix would have a positive angle in this convention (see the diagram within Figure 6.1). The ejection fraction is calculated using $EF = \frac{V_d - V_s}{V_d}$, where V_d is the uncontracted volume (corresponding to the end of *diastole*) and V_s is the contracted volume (corresponding to the end of *systole*).

The determination of the ejection fraction follows from a calculation of the volumes of the radial wedges bounded by the long axis of the cylinders and each shell finite element. The wedge volumes are calculated by first determining the volumes of 3 component tetrahedra which form each wedge. Knowing the undeformed nodal coordinates and the deformed nodal coordinates (following band contraction) allows the tetrahedral volumes to be calculated. The code used for these calculations is given in Appendix III.

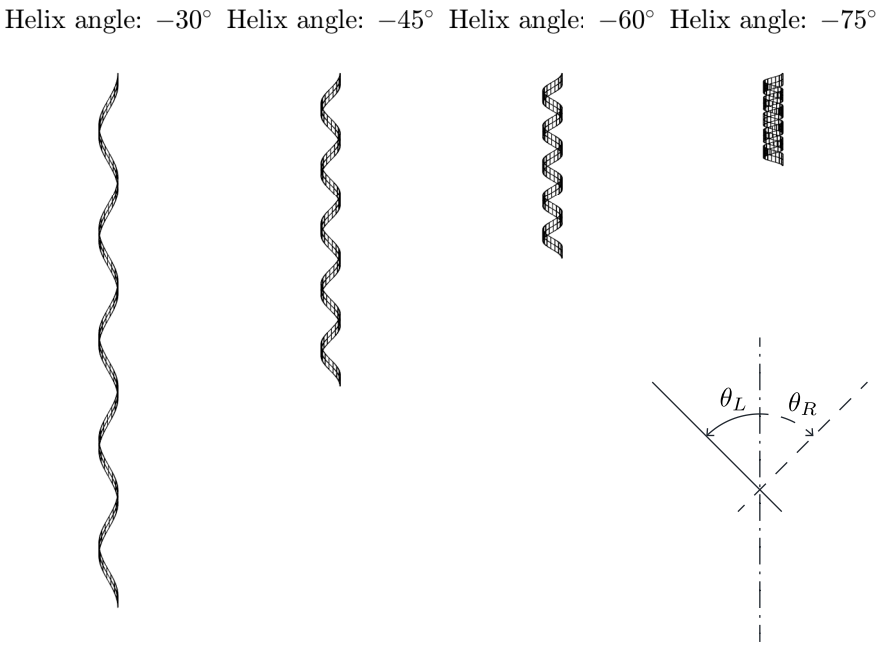


Figure 6.1: Cylindrical right helix spirals for differing helix angles (θ_R).

The same band widths and radii were used to generate the spiral cylinders shown in Figure 6.1. Note that the helix angle significantly affects the length (and openness) of each spiral.

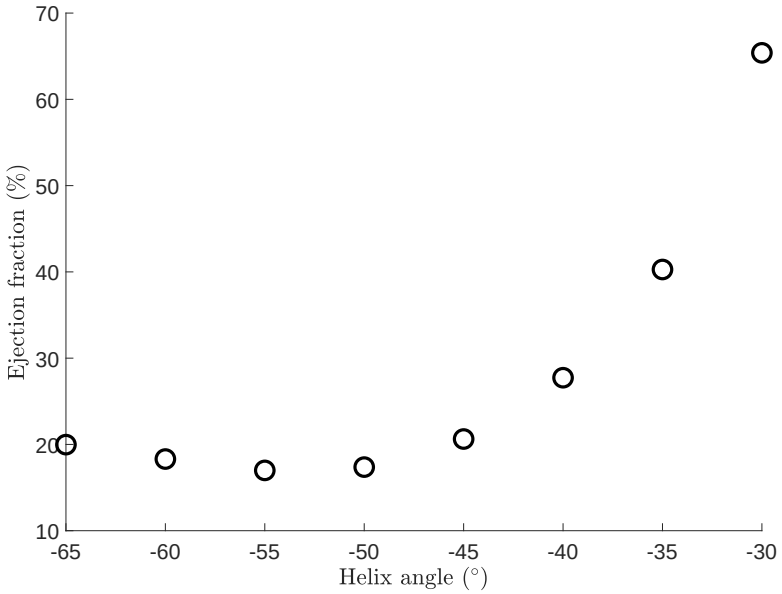


Figure 6.2: Ejection fractions for contracting spirals with differing helix angles.

Although the cylinders represent a highly simplified geometry, the findings confirm that a decrease in (the absolute value of the) helix angle produces a significantly higher ejection fraction when that angle reduces below -40°. These finite element results agree with the analytical calculations obtained by Sallin (Sallin; 1969). Note that the ejection fraction increases modestly when the helix angle decreases (that is, becomes more negative) further beyond -55°. This may due to the boundary conditions used for these analyses. The cylinders were constrained in the X, Y and Z directions at the 3 highest (largest Z coordinate) and lowest (smallest Z coordinate) nodes. Thus, in total, 6 nodes were fully restrained. Note that a total of 148 nine-noded shell elements were used for each of these analyses.

6.1.2 Double helix ventricle geometry

In all that follows, the finite element simulations are performed on the more sophisticated (but nevertheless, still simplified) representation of the left ventricle (Figure 6.3). The geometry that defines the double helix band (representing part of the left ventricle and a portion of the right ventricle) is given by the expressions developed by Grosberg (Grosberg; 2008). This model offers an idealisation of the Torrent-Guasp band (Figure 1.2). The epicardial band starts in the region of the aorta on the anterior of the basal plane and descends rapidly *down* the anterior left ventricle to the apex in a left helix. There it loops around the apex and becomes the endocardial band *climbing* up the posterior left ventricle as a right helix, forming the interventricular septum. It then wraps around the left ventricle basal plane before encircling the posterior of the right ventricle basal plane to end up close to the pulmonary artery.

Grosberg's expressions capture a single transition from left (anterior epicardium) to right (anterior endocardium) helical spirals. They do not reproduce the double left-right, left-right helical spirals of the Torrent-Guasp band. Nevertheless, with Gosberg's geometry, the band correctly skirts the ventricular surfaces by lying on one of two paraboloids (one for the left ventricle and the other for the right ventricle). The *factors* f_1 and f_2 were introduced by Grosberg to: (i) avoid her hexahedral finite element mesh occupying the same space in the (epicardium/endocardium) cross-

over region and (ii) control the curvature of the anterior surface outwards from the septum.

The circumferential angle, θ, is measured anticlockwise (viewed from the basal plane) starting on the anterior basal plane; returning to the same location after having completed 2 full turns[1].

$$0 \leq \theta \leq 4\pi \tag{6.1}$$

The normalised local coordinate $\tilde{\xi}$ expresses the relative distance across the width (*height*) of the band.

$$-1 \leq \tilde{\xi} \leq +1 \tag{6.2}$$

θ, z and r collectively define the 3 polar (cylindrical) coordinates for the myocardial band.

$$z(\theta) = C_0 - \frac{C_1.\theta^{C_2}}{\exp^{\frac{\theta}{C_3}}} + \frac{\xi.W}{2} \tag{6.3}$$

$$r(\theta, z) = f_1 f_2 \sqrt{\frac{z}{C_6} - C_7} \tag{6.4}$$

with

$$f_1 = 1 + C_4.\cos\left(\frac{\theta}{2} + \frac{\pi}{2}\left(1 - \cot\left(\frac{\pi}{C_2.C_3}\right)\right)\right) \tag{6.5}$$

$$f_2 = 1 + \frac{C_5}{\exp^{(\theta - 3.8\pi)^2}} \tag{6.6}$$

Note that an error appeared in Grosberg's original equation equivalent to (6.3), in relation to C_1. This was corrected following contact with Dr. Grosberg. The

[1]This θ is not to be confused with the helix angles θ_L or θ_R mentioned earlier.

rectangular Cartesian coordinates are obtained from

$$x(r, \theta) = r.\cos(\theta) \tag{6.7}$$

$$y(r, \theta) = r.\sin(\theta) \tag{6.8}$$

Table 6.1 lists the constants appearing in the above expressions. That Table also shows the corresponding symbols used by Grosberg and the default values used by Grosberg for each constant. Grosberg's expressions were not written in dimensionless form, thus constants C_0, C_1, C_6, C_7 and W (appearing above) all have length units associated with them and θ is expressed in radians.

	(Grosberg; 2008)	default values
C_0	C_1	4.9 cm
C_1	$C_2.A_2$	3.5 cm
C_2	A_1	1.3
C_3	B_2	2
C_4	C_4	$\frac{1}{6}$
C_5	C_5	1
C_6	A_3	1 cm^{-1}
C_7	$C_3.ZA$	0.847 cm^2
W		2.5 cm

Table 6.1: The constants used for generating the Torrent-Guasp's heart model, with a comparison against the symbols (and default values) used by Grosberg.

W defines the myocardial band width (actually the *height*). While Grosberg's band model clearly does not offer a fully-closed surface representation of the ventricles, it does capture the essential features of the geometry (in particular, the band orientation and left-right helix cross-over) providing a valuable idealisation which enables the key characteristics of the deformation (the contraction and the twisting) of the left ventricle to be simulated. In this work, Grosberg's default values are used with the exception of the band width. The band width used here was 1.25 cm (Grosberg reported using 2.5 cm) with a radial thickness of 0.5 cm. The narrower width is a consequence of the larger width giving rise to a negative discriminant appearing in (6.4) in the region of the apex. The helix angle resulting from this model varied

over the ventricles but approached 30° in the descending branch of the epicardial band. Histological studies confirm that the myocardial angles vary from 30° to 40° in the epicardium to -30° to -40° in the endocardium (see (Holzapfel and Ogden; 2009)). The boundary conditions set for both basal band ends are such that they are fixed in X and Y directions (marked with red dots in Figure 6.3). Furthermore, the apex point (the node with the lowest Z coordinate) was fixed in its Z displacement (marked with blue dot in Figure 6.3). Note that the effect of the pressure acting on the heart from the surrounding organs and thorax are not considered here. This is consistent with Grosberg's boundary condition as there is evidence to suggest the apex does not move up or down (Grosberg; 2008) (Sengupta et al.; 2008).

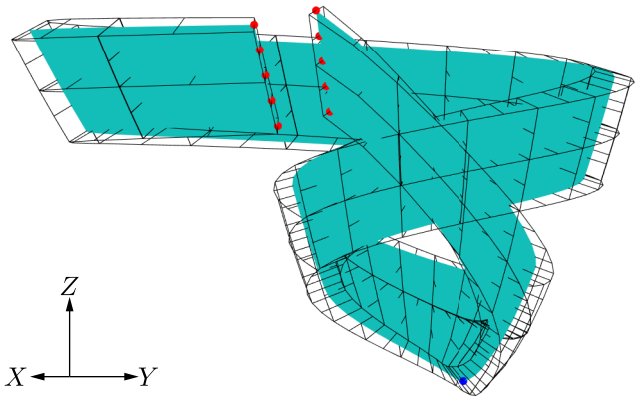


Figure 6.3: The simplified Torrent-Guasp model (view from the anterior face).

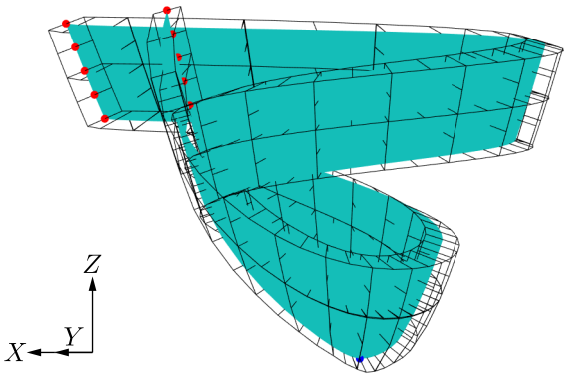


Figure 6.4: The simplified Torrent-Guasp model (view corresponding to Figure 6.3 following a 90 degree clockwise rotation).

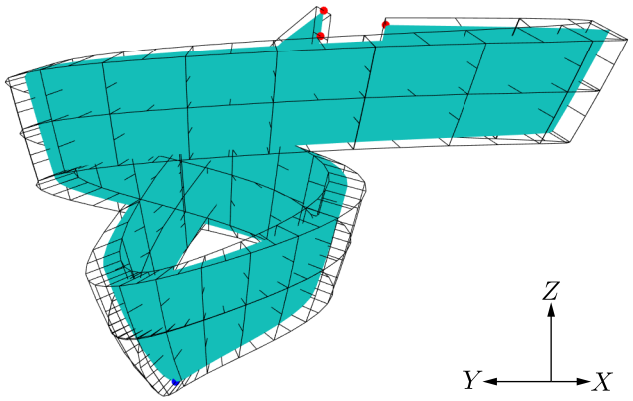


Figure 6.5: The simplified Torrent-Guasp model (view corresponding to Figure 6.3 following a 180 degree clockwise rotation).

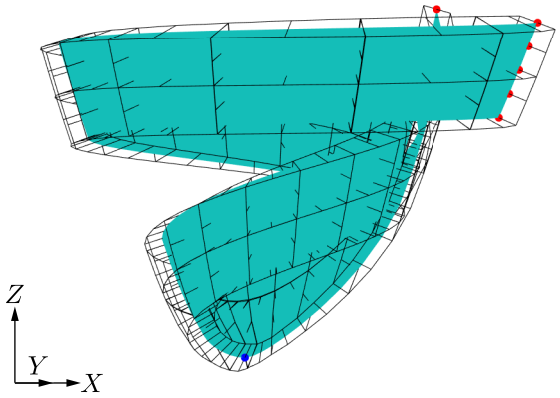


Figure 6.6: The simplified Torrent-Guasp model (view corresponding to Figure 6.3 following a 270 degree clockwise rotation).

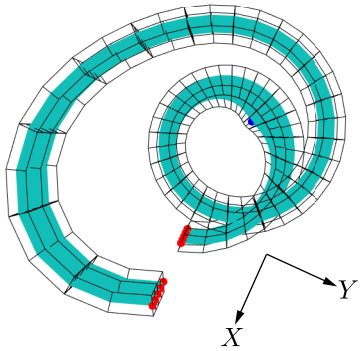


Figure 6.7: The simplified Torrent-Guasp model (view from the basal plane, showing the *inner* left ventricle and *outer* right ventricle).

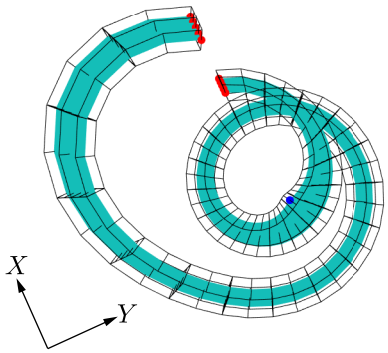


Figure 6.8: The simplified Torrent-Guasp model (view from the apex).

The effects of changing mesh density were explored. The standard case involved 64 nine-noded shell elements arranged with 32 elements over the length and 2 over the width with a $0.645ms$ time step. Subsequent analyses involves 256 and 384 elements. For all the meshes, the element sizes decreased in the regions of highest curvature (near the apex). The results are shown in Figure 6.9.

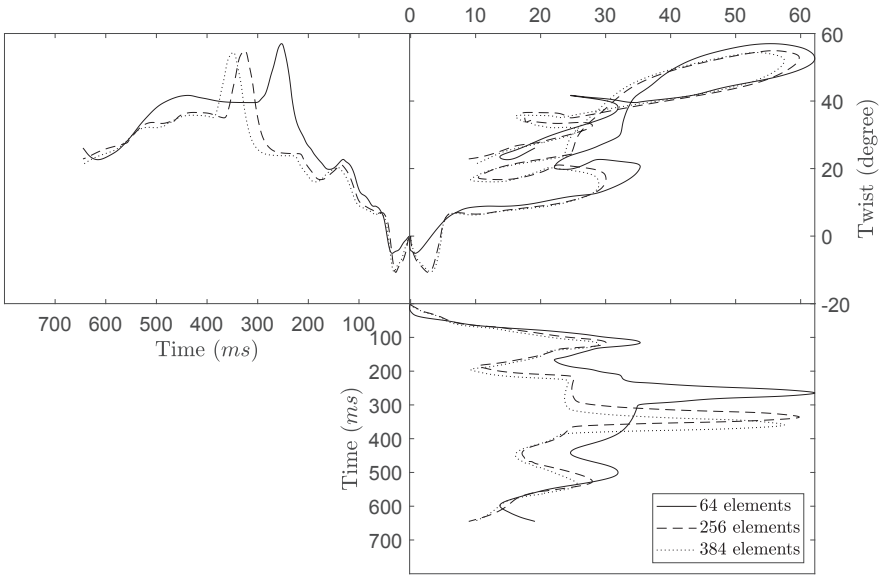


Figure 6.9: Twist against ejection fraction plot for active strain with excitation introduced at the apex for different mesh densities.

The time required for the standard case was 40 minutes. Whereas, for the 256 element analysis the run-time was 7 hours and it took 10.5 hours for the 384 element analysis. Following these analyses, it was decided that 64 elements were sufficient to capture the general trend for the simulations while providing a very significant saving on the computation cost. Therefore, the 64 element mesh has been used for

all the simulations in the remainder of this chapter.

6.2 Influence of the electrochemical excitation characteristics

The deformation of the myocardium might be expected to be influenced by the duration of the excitation, the rate of propagation of the excitation and the gradient of the excitation build-up and decline. It might be expected that the location where the contraction first begins also has an influence on the deformation. The following subsections examine each of those influences. Note that the simulations in this chapter only consider the ventricles. Thus only the S-T segment of the pulse (see Figure 2.9) will be simulated. The analyses make use of the Aliev-Panfilov model (the parameters for this are shown in Table 5.3). Like all existing *finite element* models of the heart, the analyses shown in this thesis do not consider the closed loop feedback mechanism (where the aorta detects the blood pressure to control the subsequent cardiac response). This feedback loop is something which future models will need to incorporate.

6.2.1 Active strain magnitude

As noted earlier, the maximum fibre strain that the myocardium undergoes has been observed to vary from between 15% to 20% with some recent findings indicating that the contraction could be as high as 25% locally (Smiseth et al.; 2016). Both 15% and 20% strain limits were used in the coupled electromechanical shell analysis to observe the differences in the left ventricle twist and ejection fraction. Figure 6.10 illustrates the transient twisting and evolving ejection fraction over a single heartbeat when excitation was introduced to the apex (the node with the lowest Z-coordinate), as predicted by the finite element model. The upper left plot reveals the twist versus time behaviour, while the lower right plot shows the ejection fraction versus time response. The upper right plot combines twist and ejection fraction to provide a most useful cardiac *signature* which has the potential to offer a valuable

diagnostic tool. Of particular note is the cross-over looping seen in the twist versus ejection fraction plot (see thick dashed lines in Figure 6.10). This was first correctly simulated by Grosberg (Grosberg; 2008) and it is gratifying to see this reproduced in the shell analysis prediction. In these plots a positive twist is clockwise when viewed from the basal plane. It is known that a healthy, normally functioning, human heart undergoes a clockwise twist ($\theta_A > 0$) at its apex and an anticlockwise twist ($\theta_B < 0$) at its basal plane. The net twist for the left ventricle is the sum $\theta_A + |\theta_B|$.

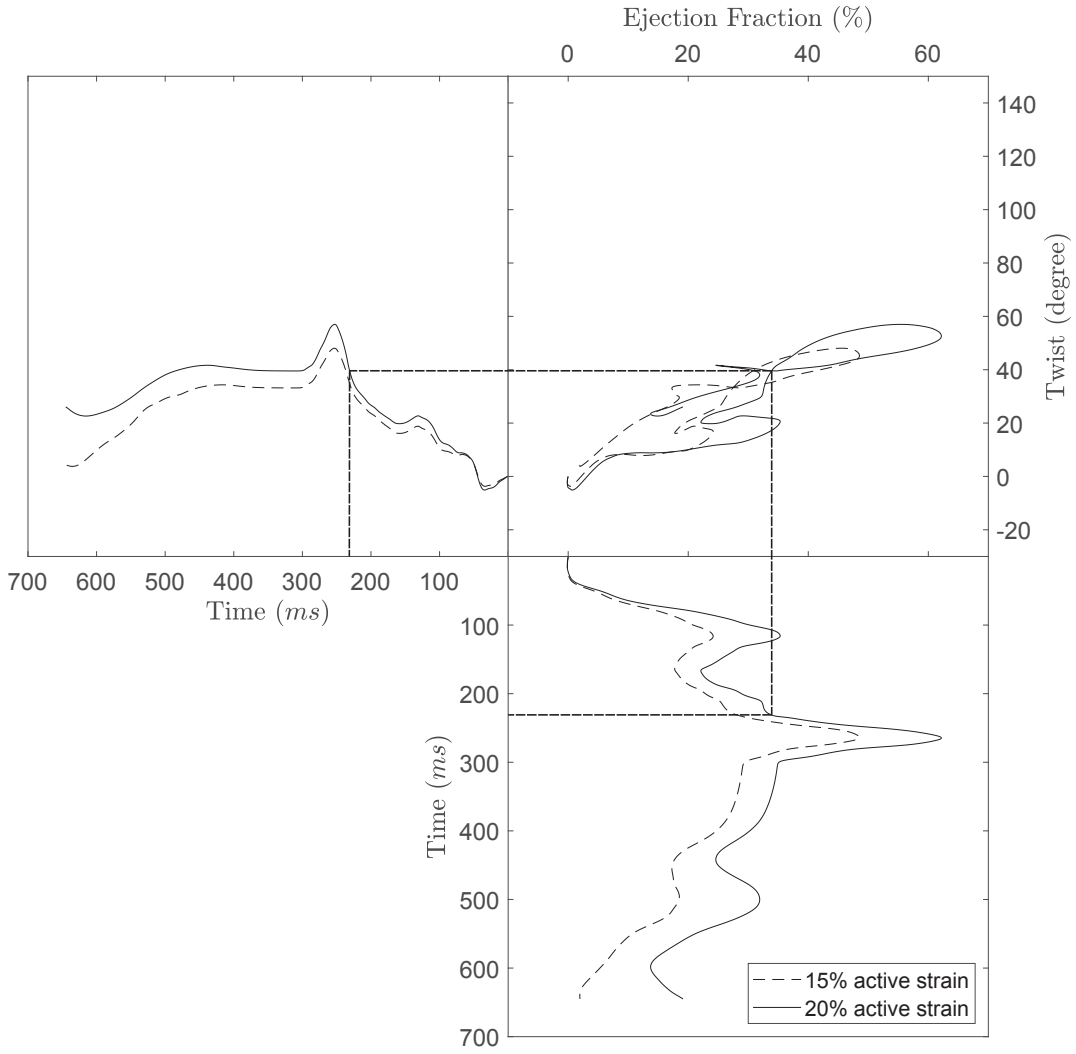
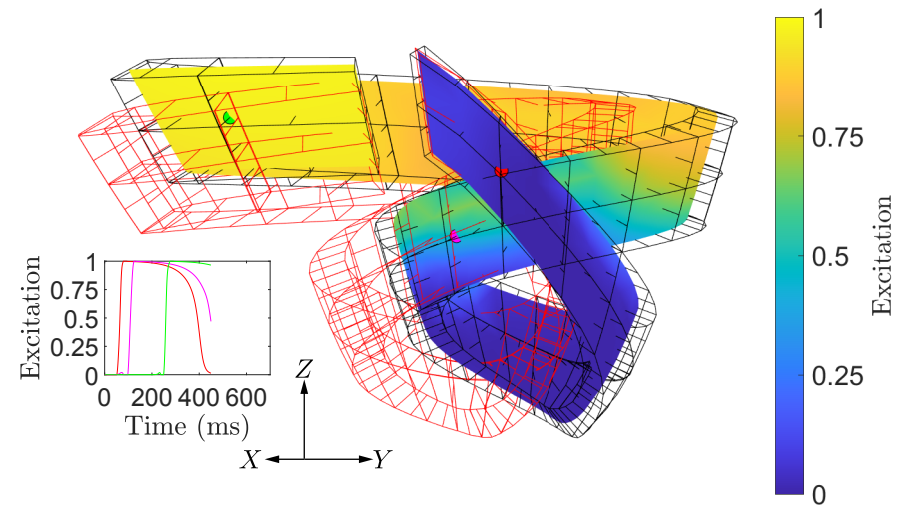


Figure 6.10: Twist against ejection fraction plot for different active strain magnitude.

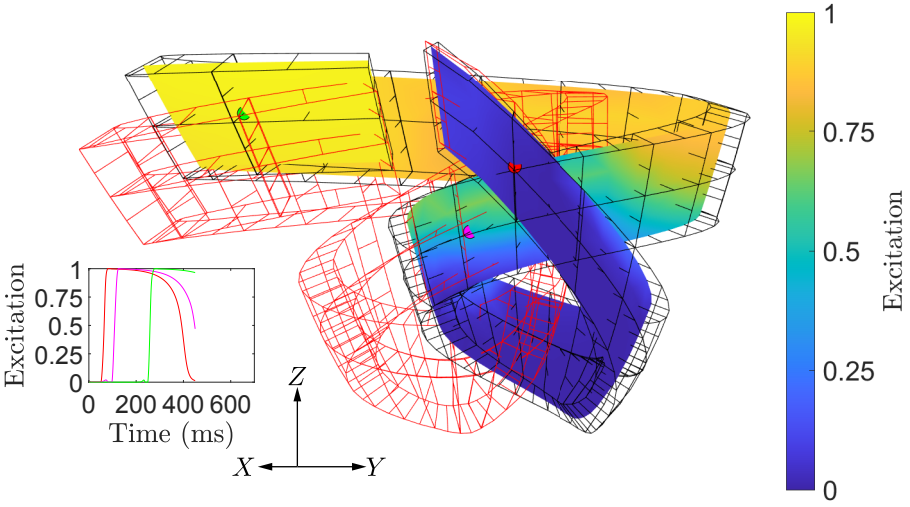
Note that the time at which the maximum twisting across coincides with the time when the maximum ejection fraction is realised (approximately 270ms for the simulation results shown in Figure 6.10). Within the medical community, it is generally recognised that the threshold of 50% is used to identify dysfunctional ($\leq$50%) and healthy (>50%) heart behaviour. Using the same geometric constants as pro-

posed by Grosberg, the 20% active strain simulation produced an ejection fraction of 62.2% whereas limiting the maximum strain to 15% gave rise to an ejection fraction of 48.5%. All subsequent simulations use an upper limit of 20% active strain. A notable difference (in addition to the higher ejection fraction) for 20% straining, is that the left ventricle remains contracted at 645ms, whereas it had relaxed at that time in the 15% strain simulation. This can be seen in Figure 6.13. Figures 6.11, 6.12 and 6.13 show the undeformed finite element mesh (black wireframe) as well as the deformed mesh (red wireframe). The degree of excitation is illustrated on the undeformed mesh; where yellow indicates an excited (contracting) region and blue indicates the region where it remains, or becomes, unexcited. Note that the large rotation of the band (for example, seen in Figure 6.12) is partly a consequence of the boundary conditions imposed whereby the ends of the myocardial band are free to move in the Z-direction while the apex is not.

The 3D nature of the myocardial band deformation makes the interpretation of a single isometric view rather difficult. It is more revealing to examine the transient rotation and ejection fraction plots, as illustrated by Figure 6.10, for example.

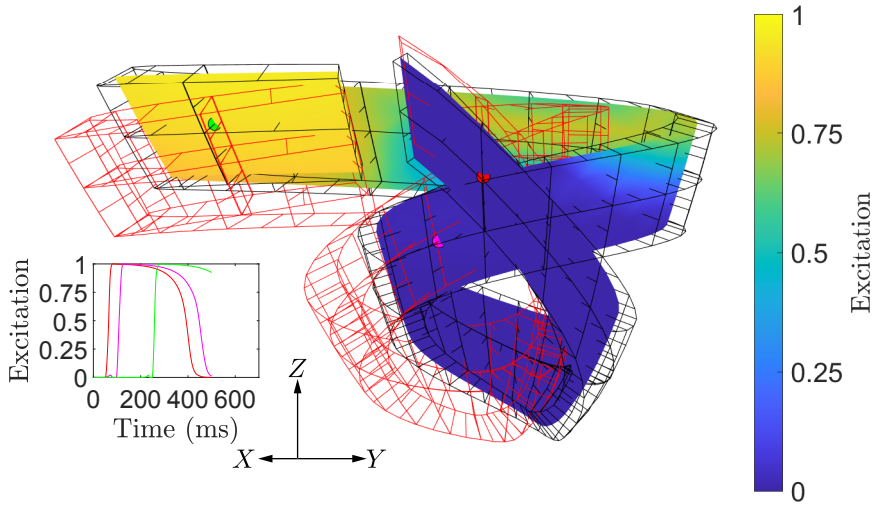


(a) 15% active strain

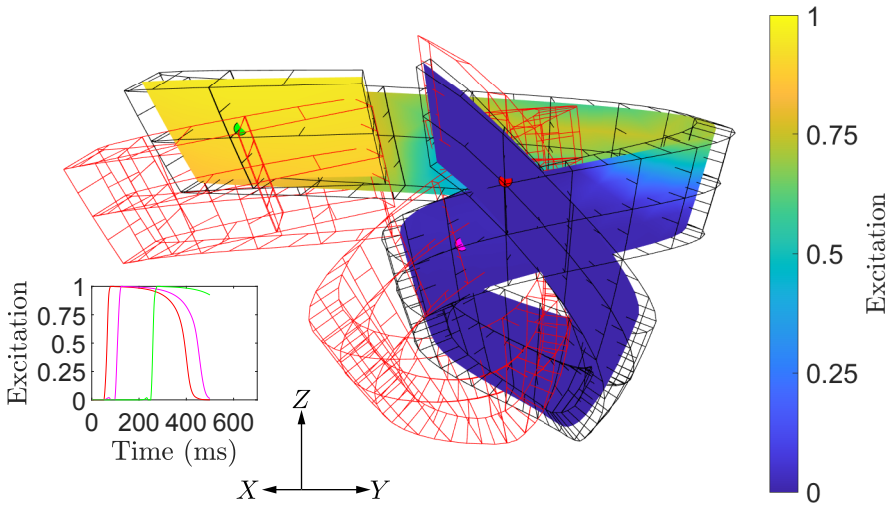


(b) 20% active strain

Figure 6.11: The initial and deformed geometry with the membrane potential distribution within the myocardium at 450ms, resulting from the coupled shell finite element analyses (15% and 20% maximum straining).

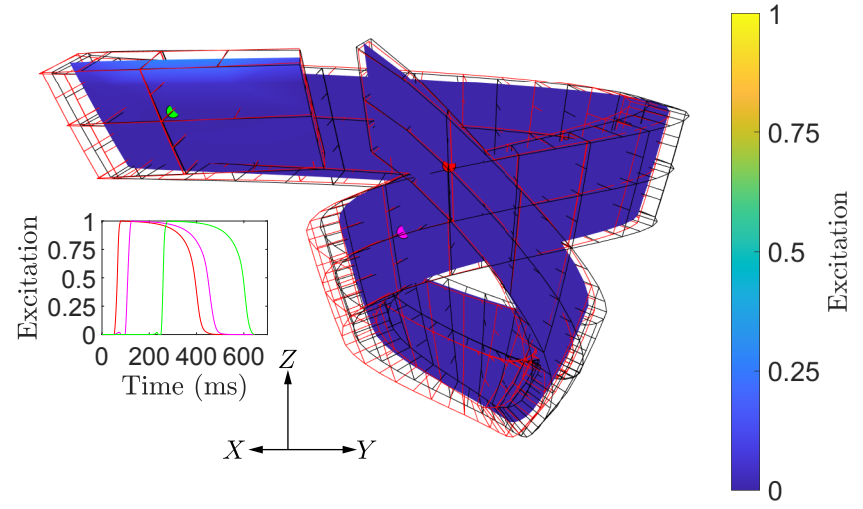


(a) 15% active strain

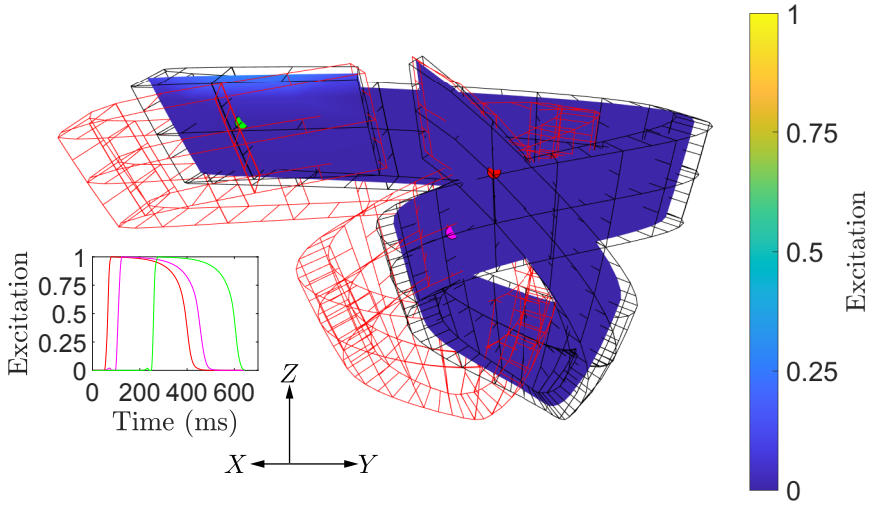


(b) 20% active strain

Figure 6.12: The initial and deformed geometry with the membrane potential distribution within the myocardium at 500ms, resulting from the coupled shell finite element analyses (15% and 20% maximum straining).



(a) 15% active strain



(b) 20% active strain

Figure 6.13: The initial and deformed geometry with the membrane potential distribution within the myocardium at 645ms, resulting from the coupled shell finite element analyses (15% and 20% maximum straining).

6.2.2 Initial excitation location

In the finite element simulations, the band excitation can be introduced at any location. In the simulations shown earlier, the excitation was introduced at the apex (lowest point in the band). This is consistent with our understanding of the sequence in which the action potential sweeps around the myocardium (see Chapter 2). The previous response is now compared with the results from a simulation where the excitation is initiated along the width of the band at the basal end of the left ventricle.

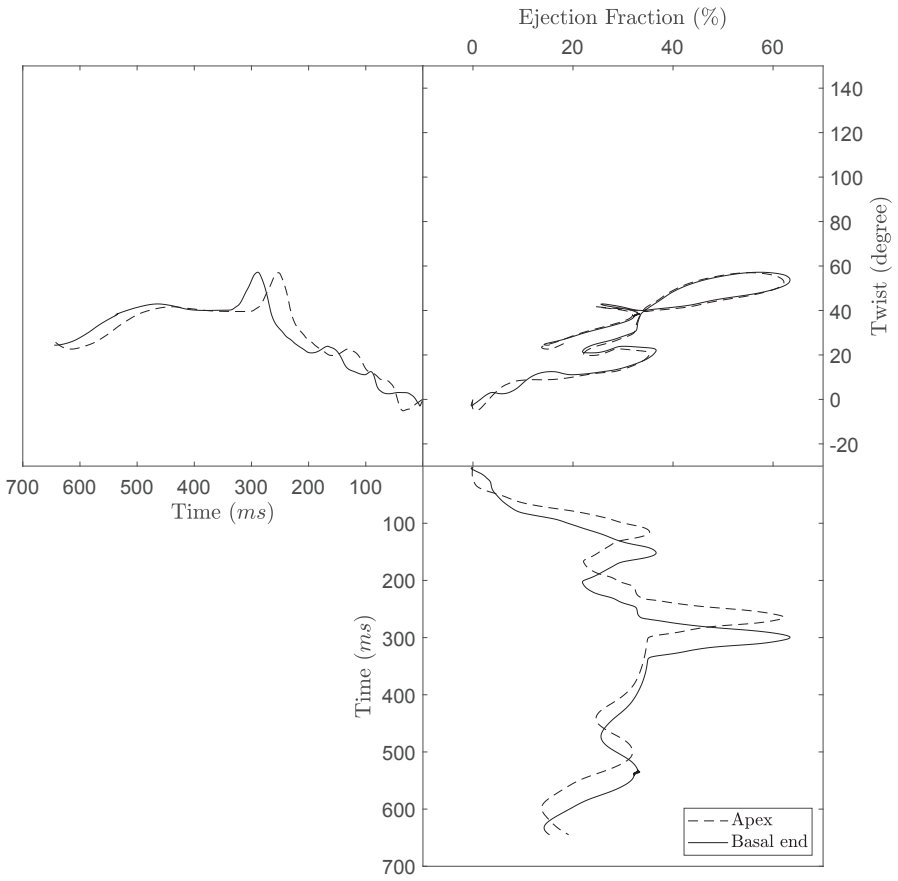
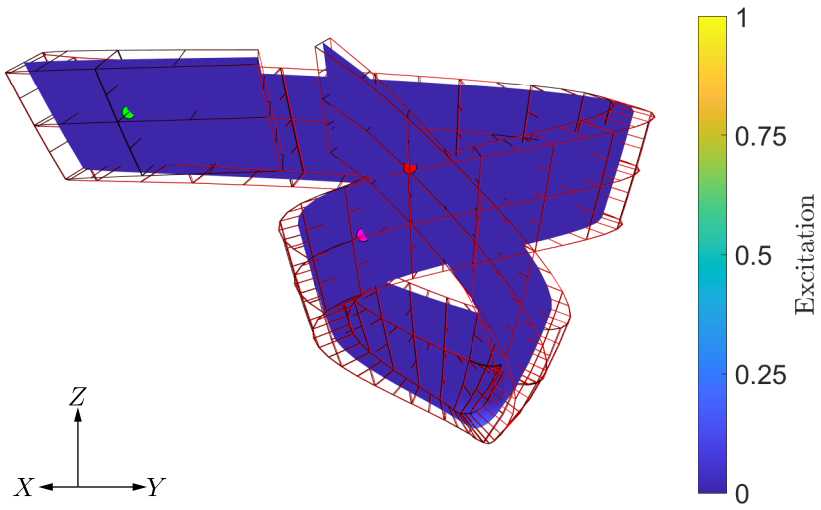
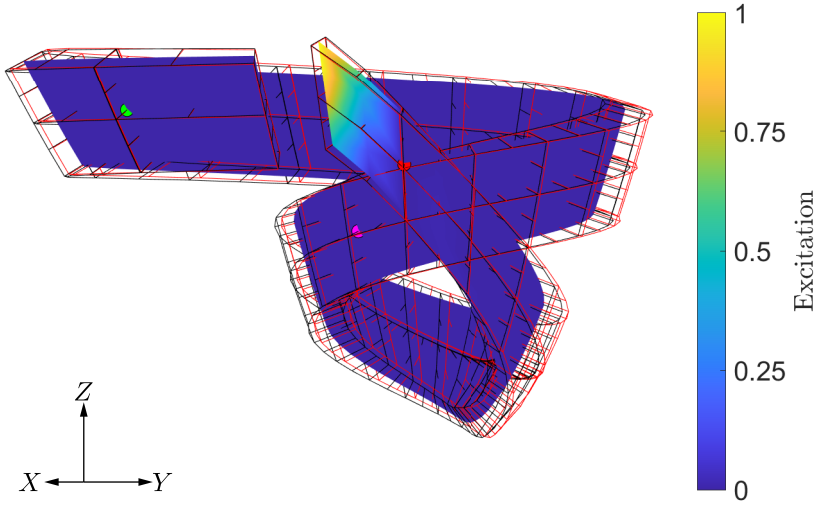


Figure 6.14: Twist against ejection fraction plot for 20% active strain with excitation introduced at the different locations.

Any differences (in terms of ejection fraction against twist) between between excitation starting in apex or basal end (see Figure 6.14) are difficult to detect. This similarity of results agrees with the findings from Grosberg (Grosberg; 2008). Somewhat surprisingly, the starting location of the excitation wave seems to play a minor role in the overall behaviour associated with myocardial contraction. However, upon reflection, this can perhaps be seen as a consequence of the excitation of the cardiac muscles lasting 400ms while the whole contraction and relaxation cycle lasts only 600-700ms. This fact (together with the excitation propagation rate) results in a significant portion of the myocardium being excited at the same time (see Figure 6.16). That is, the excitation appears to not be highly localised. A more detailed representation of the left ventricle could provide further insights into this finding.

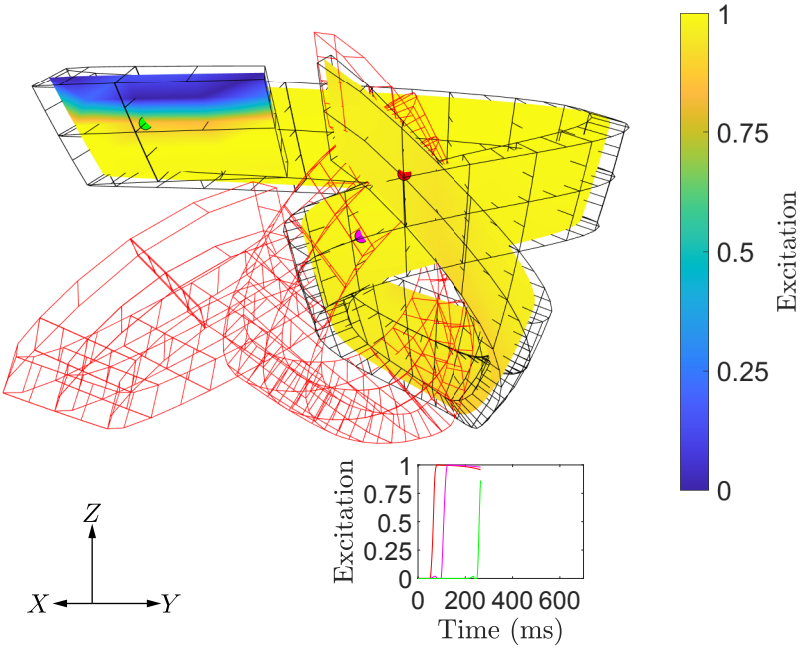


(a) Starting excitation location at the apex. The light blue region is just visible at the very end of the apex.

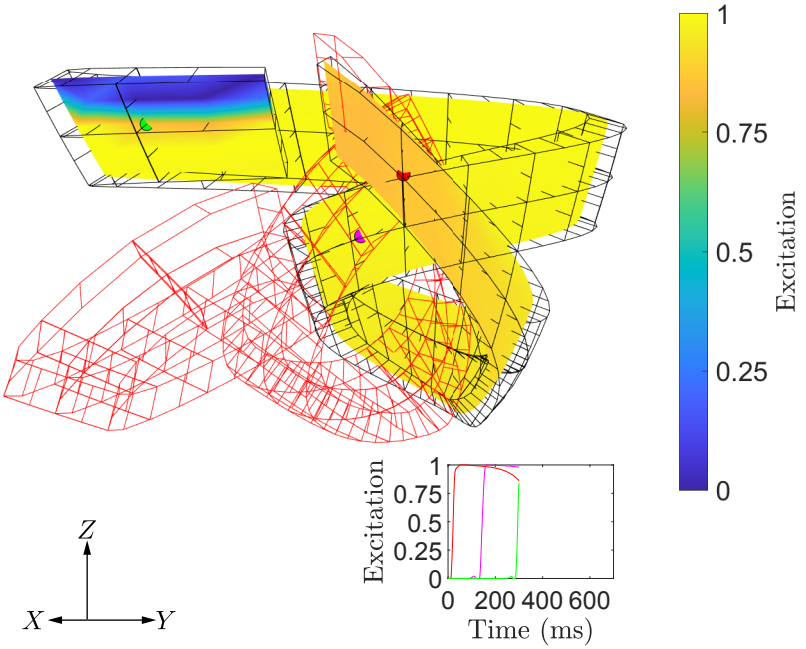


(b) Starting excitation location at the basal end.

Figure 6.15: Starting excitation locations for the apex and basal end cases.



(a) Peak contraction condition for excitation starting at the apex.



(b) Peak contraction condition for excitation starting at the basal end.

Figure 6.16: Excitation distribution during the peak contraction for apex and basal end starting condition at 265ms and 300ms respectively.

6.2.3 Electro-diffusivity

Simoons and Hugenholtz (Simoons and Hugenholtz; 1975) recorded that the ST segment of an ECG recording (see Figure 2.9, Chapter 2) shortens when an individual

is undergoing exercise. This shows that the cardiac muscles will begin relaxing faster as the heart rate is increased. To explore this, the diffusivity rate of the membrane potential was set to half and double that used in the previous simulations.

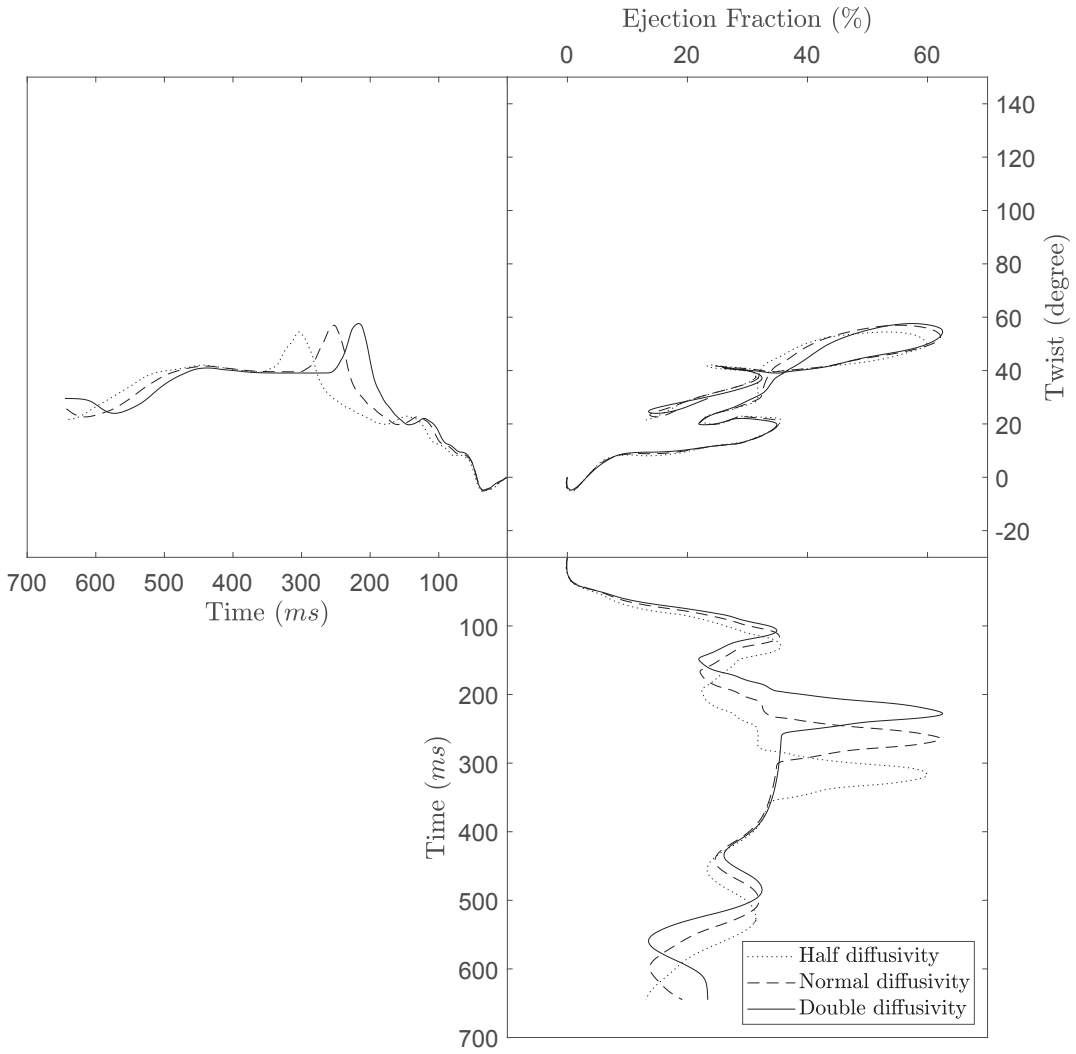


Figure 6.17: Twist against ejection fraction plot for 20% active strain with different membrane potential diffusivity.

The maximum ejection fraction for the lower diffusivity was 60%, whereas a higher diffusivity resulted in an ejection fraction of 62.5% (with the standard model used earlier producing 62.2%). Although the changes are very modest, these results agree with our expectations as a higher diffusivity allows the membrane potential to propagate faster, to cover a greater area and result in more regions of the heart undergoing contraction at the same time. Furthermore, while the ejection fractions are similar, the twist-ejection fraction plots exhibit quite different signatures. Thus far, no experimental data have been reported which illustrate the twist versus ejec-

tion fraction relationships. However, advances in imaging techniques are very likely to produce such plots in the near future.

6.2.4 Excitation duration

The other parameter that could affect the shortening of the ST segment is a reduction in the excitation duration. This effect is explored by comparing the difference between a shorter excitation duration (200ms) against the standard excitation for a healthy relaxed adult (400ms).

Parameter	200ms	400ms
c	8	8
α	0.1	0.05
γ	0.005	0.002
μ_1	0.35	0.2
μ_2	0.4	0.3
B	0.15	0.15

Table 6.2: Input parameters for different excitation durations using the Aliev-Panfilov model.

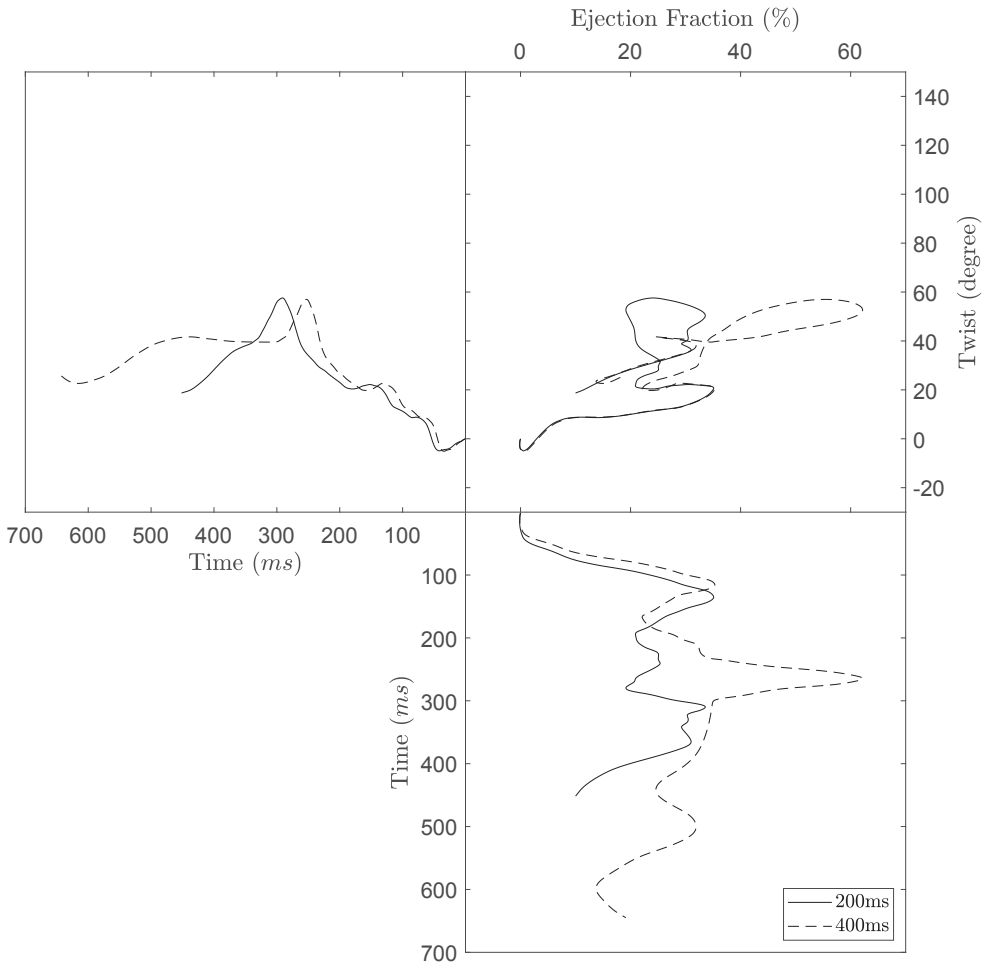


Figure 6.18: Twist against ejection fraction plot for different excitation duration with the excitation starting at the apex.

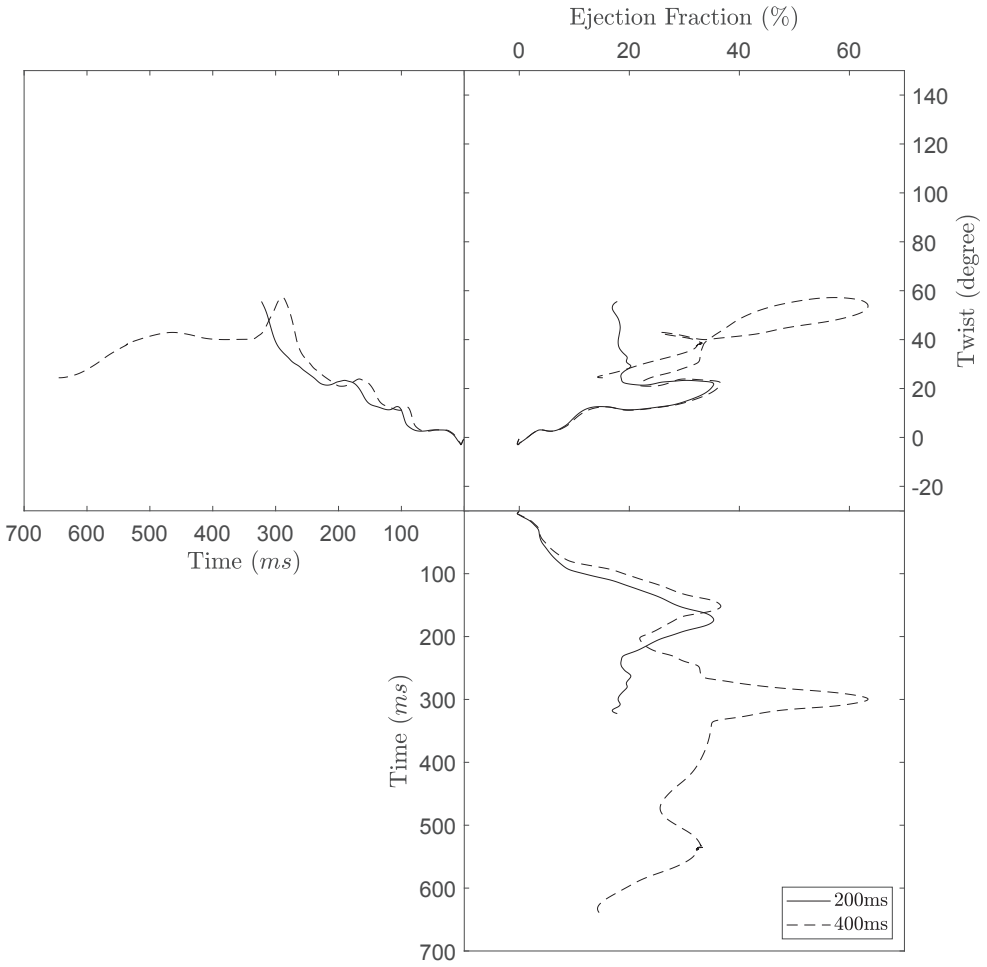


Figure 6.19: Twist against ejection fraction plot for different excitation duration with the excitation starting at the basal end.

Having a excitation of 200ms dramatically reduces the maximum ejection fraction to just 35.1%. This result agrees with the hypothesis that a longer duration of excitation allow more sections of the cardiac band to contract, leading to a higher ejection fraction. In order to achieve an ejection fraction of above 50%, the propagation speed would need to be increased. Further analysis was undertaken where a 200ms duration excitation was initiated at the basal (rather than the apical) end. This gave rise to less than 0.5% difference in the maximum ejection fraction but very different ejection fraction and twisting transients (see Figures 6.18 and 6.19). However, it should be noted that these simulations experienced convergence difficulties after 451ms and 322ms respectively, thus the analyses were curtailed at those times. These cases are examined again in Section 6.3.1.

6.3 The consequences of changes to the end diastole ventricle shape

Many cardiac weaknesses are associated with evolving heart deformities (or geometric irregularities at birth). One such condition is known as dilated cardiomyopathy. Some preliminary studies into the consequences of changes in the heart geometry are explored in Section 6.3.2. Pior to that, an investigation into the effect of introducing material in-homogeneity is undertaken.

6.3.1 Material properties of the myocardium

Within conventional finite element stress analyses, the material model (see $[D]$ in (4.30)) pays a crucial role in influencing the deformation when a solid is exposed to loading. In this study, it was originally intended to make use of the advanced anisotropic hyperelastic model of Holzapfel and Ogden (Holzapfel and Ogden; 2009) to represent the myocardium. However, in this work, the deformation has been directly imposed through the *active strain* approach, circumventing the need (at this stage) for a sophisticated material model. Thus, for these simulations the stiffness of the cardiac muscle (provided it is homogeneous throughout) has no influence on the degree of contraction. Nevertheless, changes in the relative stiffness within the cardiac muscle will have an influence and so this is now analysed.

McGarvey (McGarvey et al.; 2015) artificially introduced an infarct into porcine hearts by blocking the arteries. The infarct tissue was reported to be 20 times stiffer than the healthy tissue. Half of the left ventricle was damaged by the ligation in these experiments. This condition was replicated in the FE simulations (setting the material to be stiffer from $\theta = 0$ to $\theta = \pi$). The simulation trends produced agree with the findings from the experiments. That is, there was a significant reduction in the maximum ejection fraction (21.2%) when a stiffer zone was introduced. However, the FE simulation experienced convergence difficulties beyond 483ms.

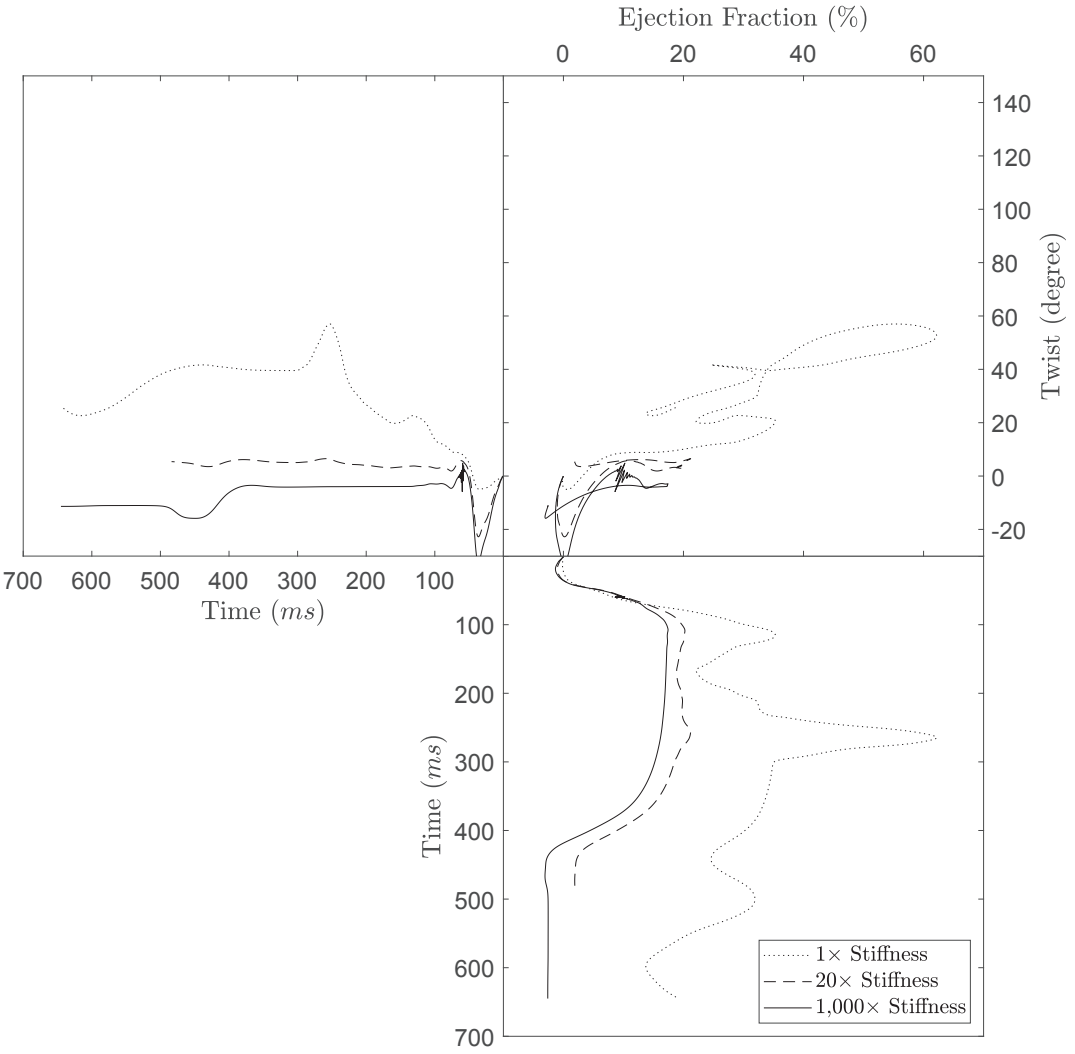
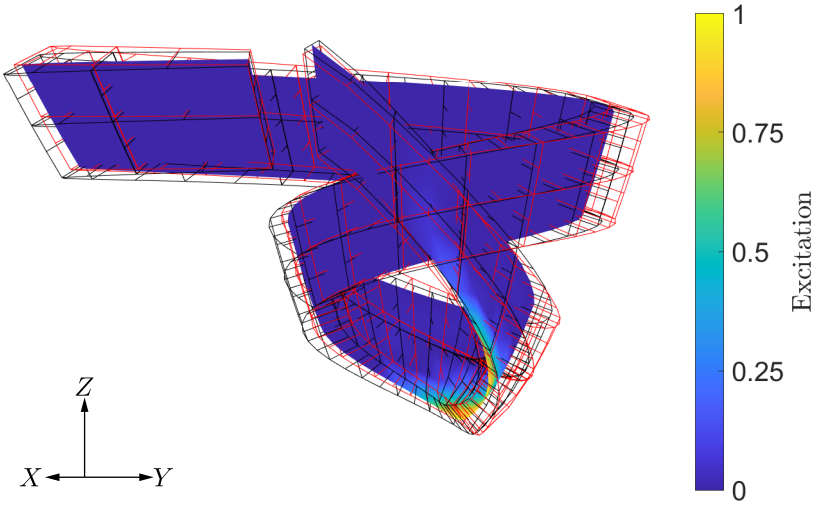
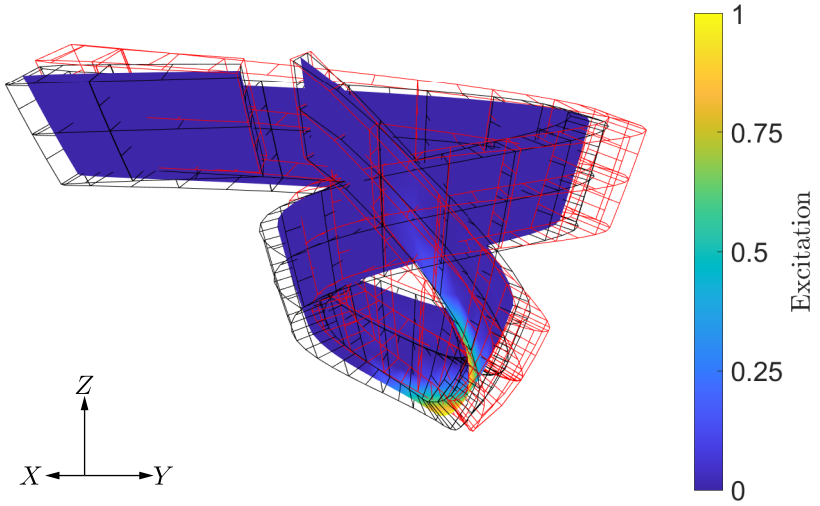


Figure 6.20: Twist against ejection fraction plot with half of left ventricle consists of different stiffness than the rest of the band (excitation starting at apex).

A noticeable feature of having half of the left ventricle stiffer was that the twist direction reversed and the ejection fraction plot showed a single elongated peak (Figure 6.20) instead of the more distinct higher peak; see (for example) Figure 6.10. During a number of the FE simulations (when using Grosberg's geometry) it was found that the very high curvature of the apex region sometimes led to convergence difficulties as the membrane potential spread through this region causing highly localised deformations. This could be overcome by: (i) creating a smoother geometry near the apex, (ii) having a more refined mesh around the apex or (iii) distributing the initial excitation over a wider area. In the work reported here, the membrane potential was spread across the full band width in the apex region to help resolve this issue (Figure 6.22).



(a) Homogeneous stiffness across the muscle band.



(b) 20 times stiffer muscle on half of the left ventricle.

Figure 6.21: Current and deformed configurations with the excitation distribution for homogeneous and heterogeneous bands at 30ms.

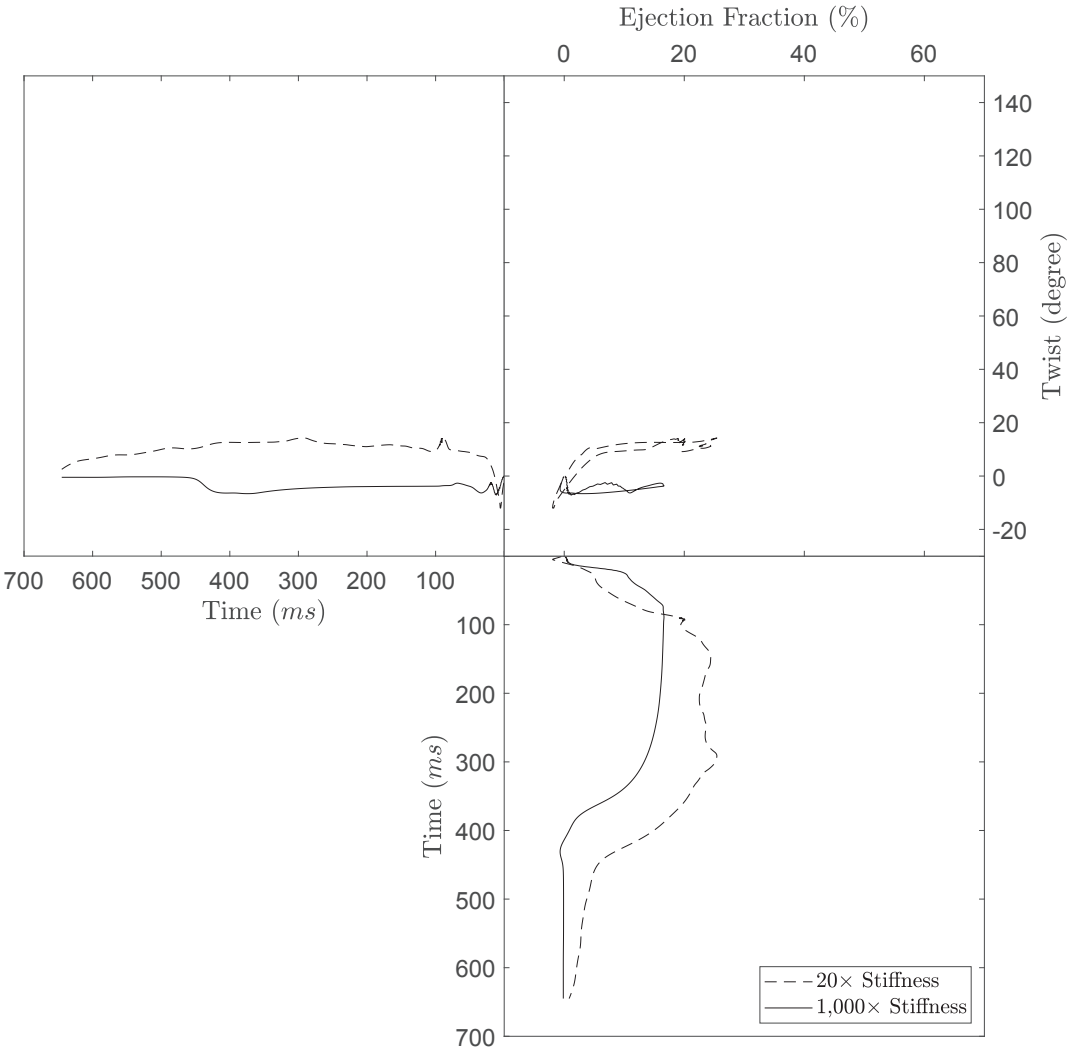


Figure 6.22: Twist against ejection fraction plot with half of left ventricle having different stiffness than the rest of the band (excitation starting across the full band width in the apex region).

Having a wider spread of initial excitation distribution did overcome the convergence difficulties, allowing the simulation to convergence throughout all 1,000 *load steps* (645ms); producing an ejection fraction of 25.4%. It was deemed interesting to explore this local stiffening further by making the infarct tissue 1,000 times stiffer (rather than 20 times stiffer).

This gave rise to an even greater reduction in the ejection fraction of 17.4% when the excitation was introduced at the apex and 16.6% when the excitation was introduced across the width of the band at the apex. Both stiffened simulations reached full convergence in all 1,000 *load steps* (645ms), whereas earlier (with the excitation introduced at the apex) simulations suffered convergence difficulties beyond 484ms.

Recognising that a stiffer material and lower deformation helps with the simulation stability, the different excitation duration cases (Section 6.2.4) were run again with half of the left ventricle being 1,000 times stiffer.

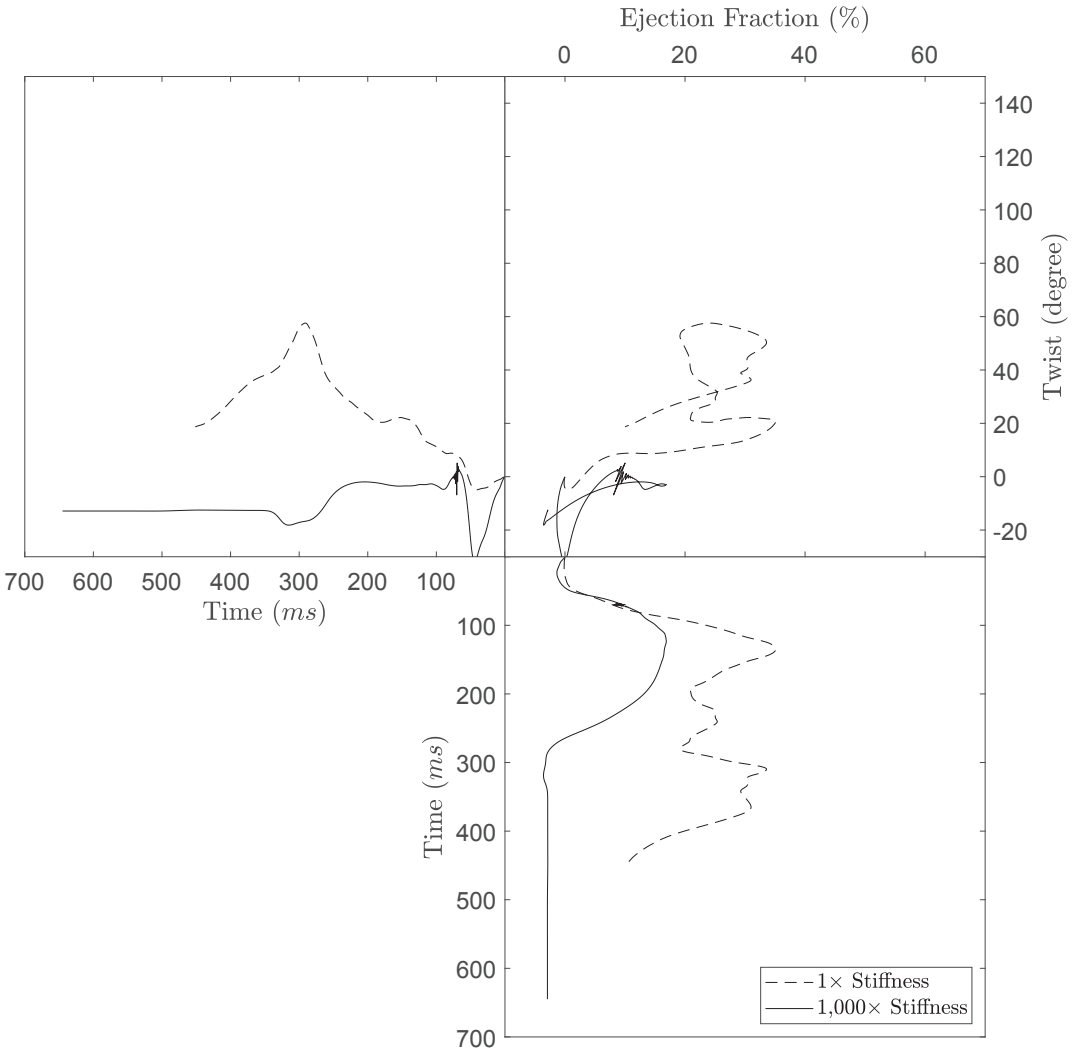


Figure 6.23: Twist against ejection fraction plot with half of left ventricle having different stiffness than the rest of the band and an excitation that lasts 200ms (excitation starting at the apex).

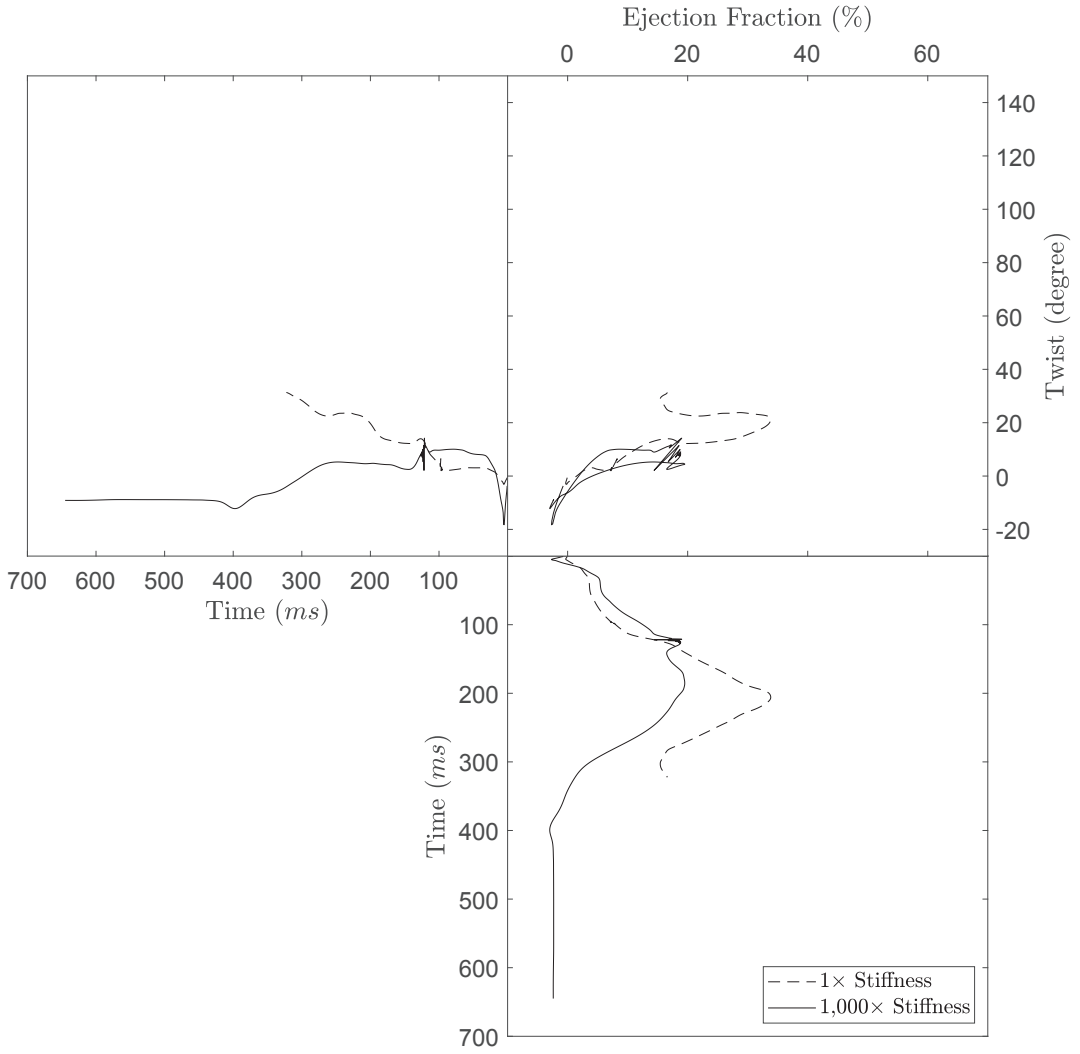


Figure 6.24: Twist against ejection fraction plot with half of left ventricle having different stiffness than the rest of the band and an excitation that lasts 200ms (excitation starting at the basal end).

In these new analyses, it was found that both cases were able to converge throughout all 1,000 *load steps*. Introducing different locations for the initiation of excitation produced 16.9% and 22.1% ejection fractions for apex and basal initiations respectively. It is also noted that when both elements forming the boundary between the stiff and soft material were fully excited, the *Gauss point* strains (in the direction of the band) in the stiffer material were 18% smaller than those recorded in the softer material. Whereas, the *Gauss point* stresses (in the same direction) were 47% higher in the stiffer material than those seen in the soft material. These findings are consistent with what is expected for this band model.

6.3.2 Macrostructure of the heart

There are 2 general deformities that might characterise an unhealthy heart: a partial collapse of the ventricle (resulting in a reduced ventricle volume) or a dilation of the ventricle muscles (resulting in an increased ventricle volume and increase in the helical fibre angle). These defects are referred to as *oblique* and *dilated* respectively and can be seen in Figure 6.26. Note that the latter case is also referred to as an increase in sphericity. Neither the *oblique* nor *dilated* heart models capture the apex region. In the case of the dilated heart the helix angle is significantly larger (more than 60°) than seen in the standard model (approximately 30°, Figure 6.25). While further work is required to represent the *oblique* and *dilated* hearts in a more realistic way, these geometries were used as they had been provided by Grosberg.

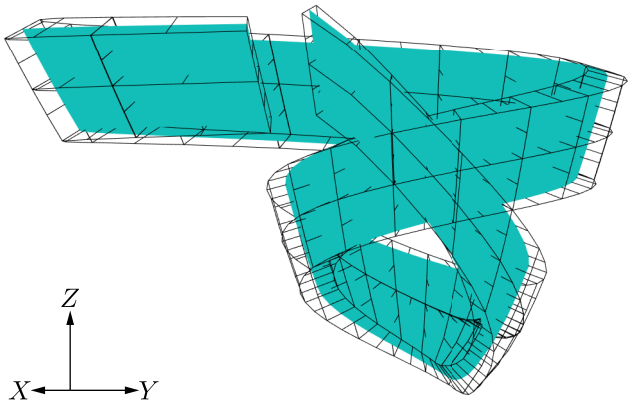
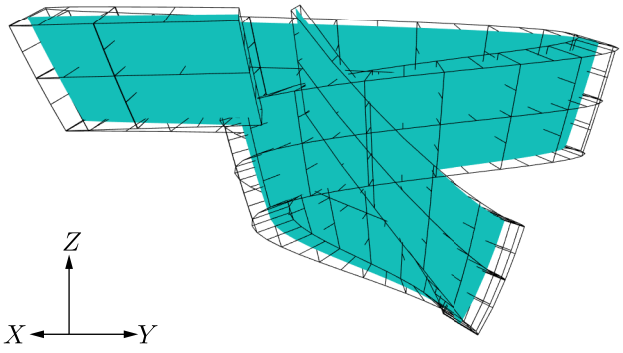


Figure 6.25: FE mesh for the standard Grosberg band model used in this work.

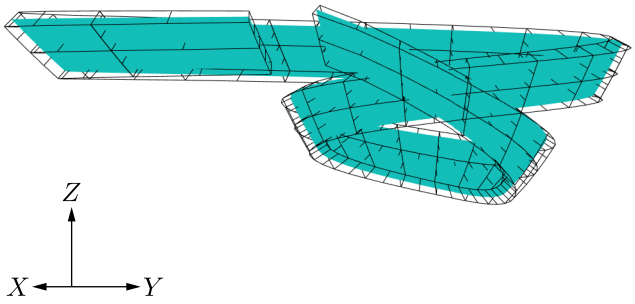
The constants used to describe the *oblique* and *dilated* hearts are given in Table 6.3.

	Oblique	Dilated
C_0	4.9 cm	4.9 cm
C_1	3.5 cm	3.5 cm
C_2	1.0	1.3
C_3	2	2
C_4	$\frac{1}{6}$	$\frac{1}{6}$
C_5	1	1
C_6	1 cm^{-1}	0.25 cm^{-1}
C_7	1.232 cm^2	0.847 cm^2
W	1.25 cm	1.25 cm

Table 6.3: The constants used for generating Grosberg's geometry for the *oblique* and *dilated* models.



(a) *Oblique* heart mesh.



(b) *Dilated* heart mesh.

Figure 6.26: FE meshes associated with the geometries of the two different myocardial band models.

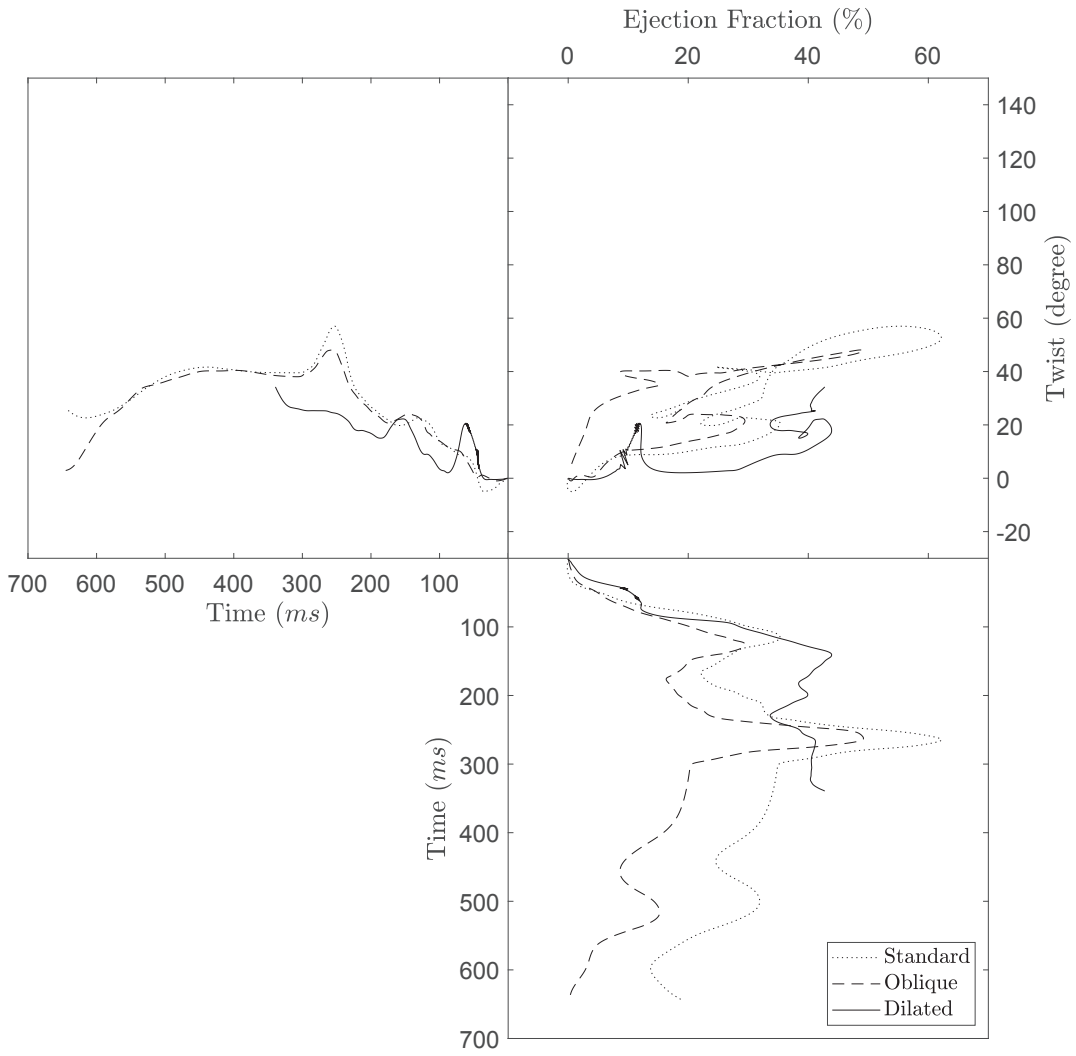


Figure 6.27: Twist against ejection fraction plot for the different heart models.

It was noted that with the geometry representing the *dilated* heart, the simulation experienced a lack of convergence beyond 322ms. The cause of this is not clear, but the twist transient exhibits an unexpected double peak. Applying the strategy of making half of the left ventricle 1,000 times stiffer enabled the simulation to converge for the entire 645ms. The twist-ejection fraction signatures are very different from those obtained from the standard model.

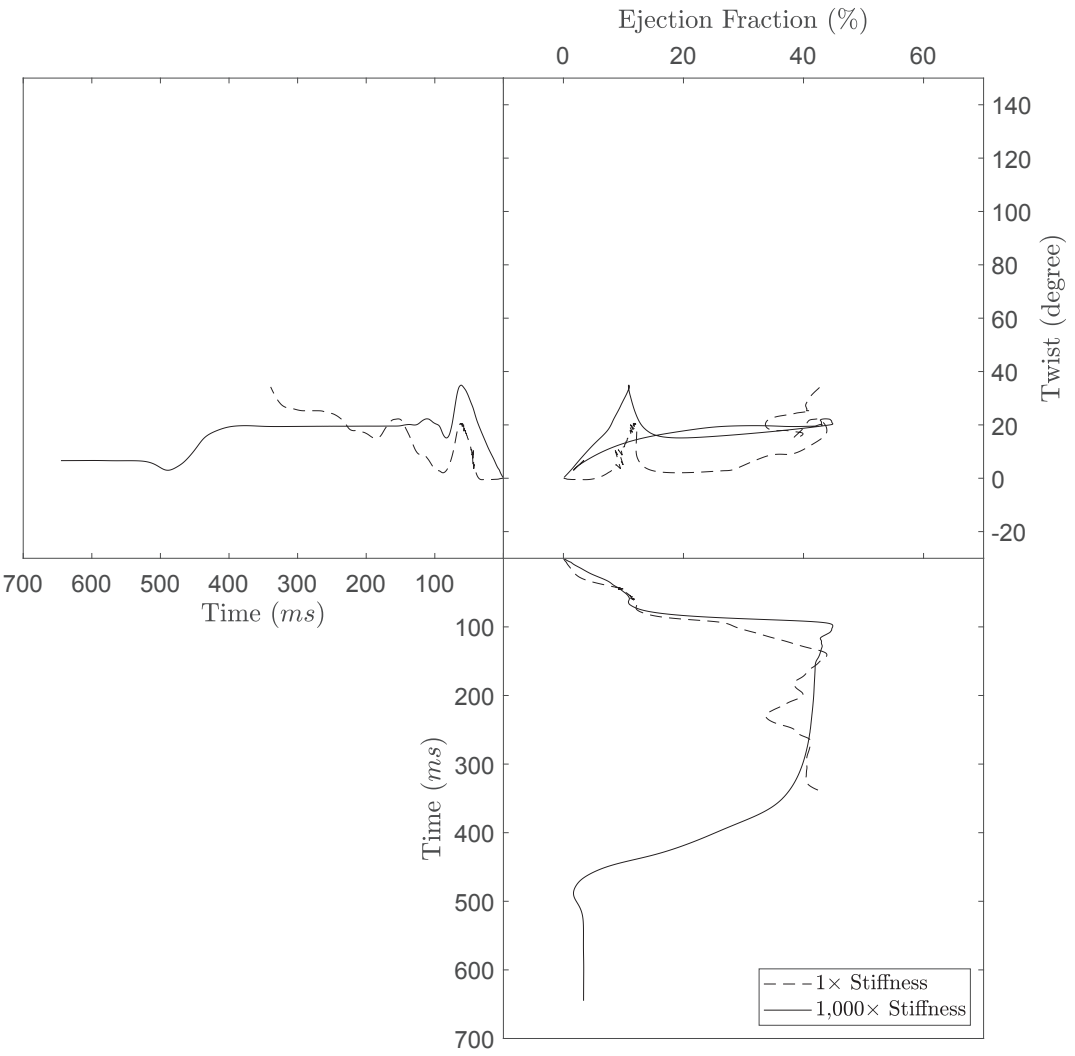


Figure 6.28: Twist against ejection fraction plot for the *dilated* heart model, having different stiffness than the remainder of the band.

It became evident that both the ejection fraction and twist calculations are sensitive to the locations where they are determined. The twist measurements reported in this chapter were gathered from the node located at $\theta = \frac{3\pi}{2}$ (as shown in Figure 6.29). The first and last node at either end of the myocardial band are fixed, so the first free node occurs at $\theta = \frac{\pi}{16}$. Twist against ejection fraction plots were produced for the standard model at 4 other locations ($\theta = \frac{\pi}{16}$, $\frac{\pi}{2}$, π and 2π) as illustrated in Figure 6.30.

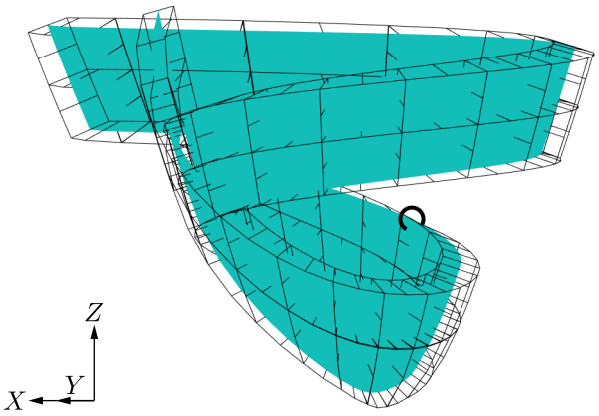


Figure 6.29: Identification of the node (at $\theta = \frac{3\pi}{2}$) used to determine the twist magnitude (relative to its original polar angle θ).

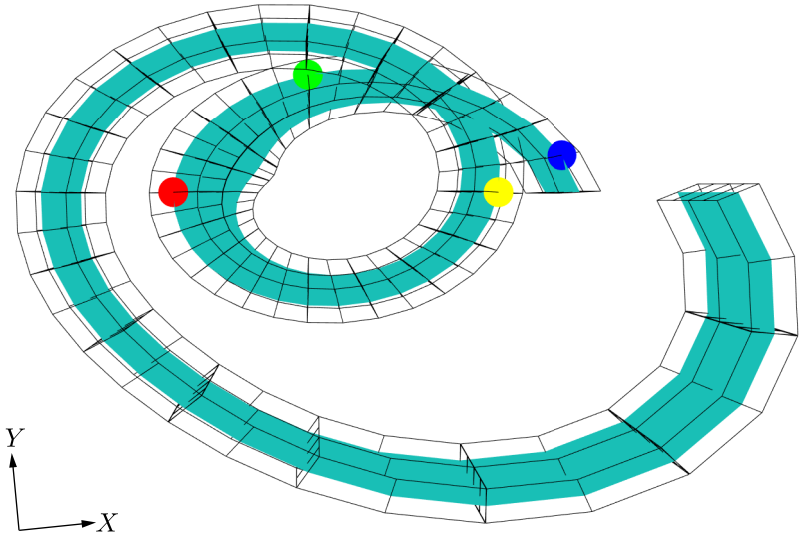


Figure 6.30: Plan view from the basal plane of the myocardial band with twist measurement nodes located at $\theta = \frac{\pi}{16}$, $\frac{\pi}{2}$, π and 2π (illustrated in blue, green, red and yellow respectively).

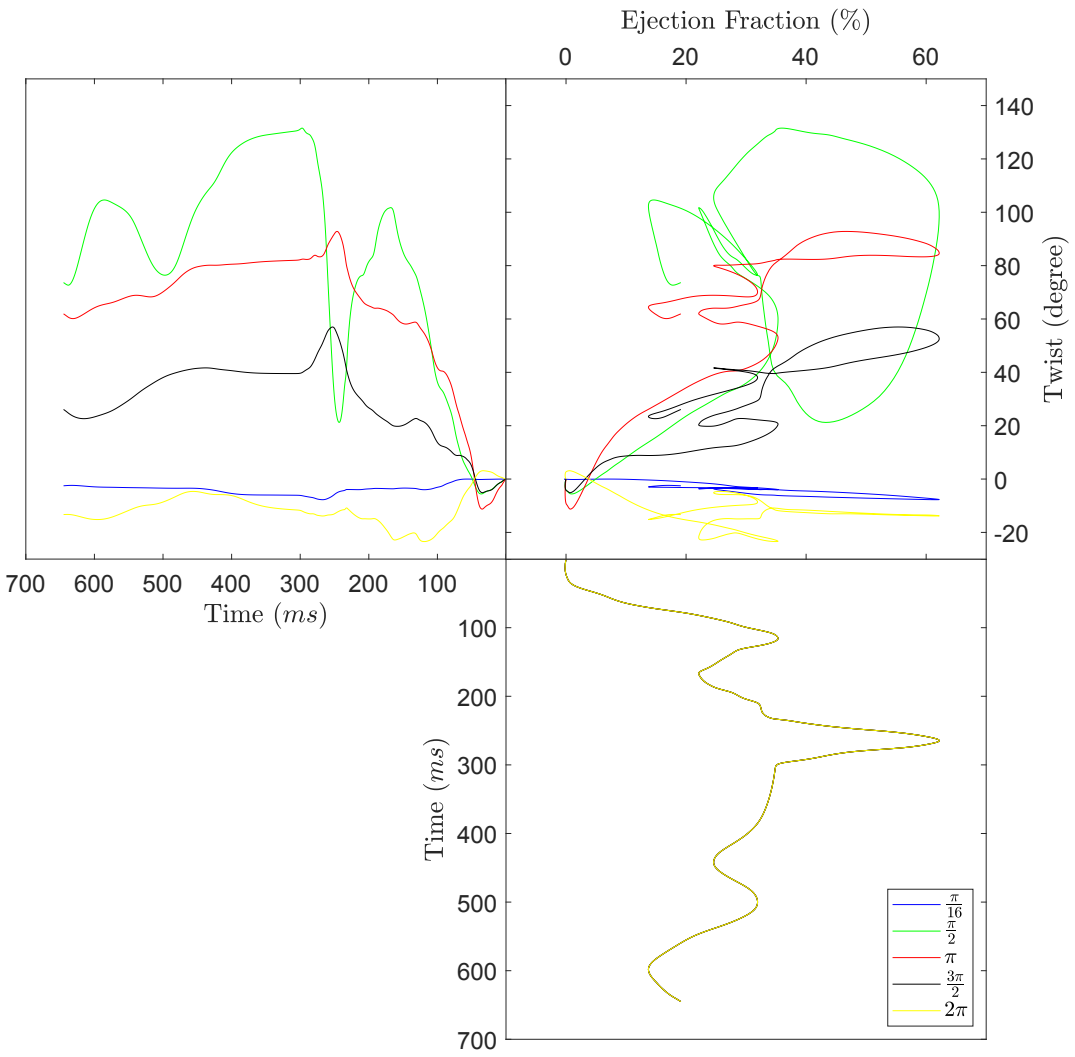


Figure 6.31: Twist against ejection fraction plot for the standard case with data from the node at different locations.

Unsurprisingly, the twists recorded at different locations are very different. The medical community needs to adopt a single agreed methodology for both measures (twist and ejection fraction) before trully meaningful comparisons can be made (Badano and Muraru; 2019). The work reported in this thesis decided to make use of the node located at $\theta = \frac{3\pi}{2}$ partly because it corresponded to a key location near the apex (but not at the apex) and partly because the twist against ejection fraction graph obtained for the standard case was similar to that given by Grosberg (Grosberg; 2008).

This Chapter has demonstrated that the novel large deformation coupled electromechanical shell finite element analysis code offers an exciting new tool with which

to explore the deformation of the myocardial band during the cardiac cycle. The availability of this numerical model invites many additional analyses and sensitivity studies to be undertaken. The approach followed here is inevitably an early step in delivering a complete model of the human heart. Yet it has already given rise to important observations with regard to the deformation of this complex and vital organ. It also has been shown to offer a computationally efficient alternative to the tetrathedral FE representations used by other researchers (Baillargeon et al.; 2014) (Hurtado et al.; 2016) (Nemavhola; 2019). Furthermore, rather than relying on commercial FE software, this study has developed and provided full MATLAB codes for all of the analyses.

References

Badano, L. P. and Muraru, D. (2019). Editorial: Twist mechanics of the left ventricle, *Circulation: Cardiovascular Imaging* **12(4)**.

Baillargeon, B., Rebelo, N., Fox, D. D., Taylor, R. L. and Kuhl, E. (2014). The living heart project: A robust and integrative simulator for human heart function, *European Journal of Mechanics / A Solids* **48**: 38–47.

Buckberg, G. D., Nanda, N. C., Nguyen, C. and Kocica, M. J. (2018). What is the heart? anatomy, function, pathophysiology, and misconceptions, *Journal of Cardiovascular Development and Disease* **5(2)**.

Göktepe, S., Acharya, S. N. S., Wong, J. and Kuhl, E. (2011). Computational modeling of passive myocardium, *International Journay for Numerical Methods in Biomedical Engineering* **27(1)**: 1–12.

Grosberg, A. (2008). *A Bio-Inspired Computational Model of Cardiac Mechanics: Pathology and Development*, PhD thesis, California Institute of Technology.

Holzapfel, G. A. and Ogden, R. W. (2009). Constitutive modelling of passive myocardium: A structurally based framework for material characterization, *Philosophical Transactions of The Royal Society A* **367**: 3445–3475.

Hurtado, D. E., Castro, S. and Gizzi, A. (2016). Computational modeling of non-linear diffusion in cardiac electrophysiology: A novel porous-medium approach, *Computer Methods in Applied Mechanics and Engineering* **300(1)**: 70–83.

Kotikanyadanam, M., Göktepe, S. and Kuhl, E. (2010). Computational modeling of electrocardiograms: A finite element approach toward cardiac excitation, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING* **26**: 524–533.

McGarvey, J. R., Mojsejenko, D., Dorsey, S. M., Nikou, A., Burdick, J. A., Gorman, J. H., Jackson, B. M., Pilla, J. J., Gorman, R. C. and Wenk, J. F. (2015). Temporal changes in infarct material properties: An in vivo assessment using magnetic resonance imaging and finite element simulations, *The Society of Thoracic Surgeons* **100(2)**: 528–529.

Nemavhola, F. (2019). Detailed structural assessment of healthy interventricular septum in the presence of remodeling infarct in the free wall - a finite element model, *Heliyon* **5(6)**.

Sallin, E. A. (1969). Fiber orientation and ejection fraction in the human left ventricle, *Biophysical Journal* **9(7)**: 954–964.

Sengupta, P. P. and Narula, J. (2014). Cardiac strain as a universal biomarker: Interpreting the sounds of uneasy heart muscle cells, *JACC: Cardiovascular Imaging* **7(5)**: 534–536.

Sengupta, P. P., Tajik, J., Chandrasekaran, K. and Khandheria, B. K. (2008). Twist mechanics of the left ventricle: Principle and application, *JACC: Cardiovascular Imaging* **1(3)**: 366–376.

Simoons, M. L. and Hugenholtz, P. (1975). Gradual changes of ecg waveform during and after exercise in normal subjects, *Circulation* **52(4)**: 570577.

Smiseth, O. A., Torp, H., Opdahl, A., Haugaa, K. H. and Urheim, S. (2016). Myocardial strain imaging: How useful is it in clinical decision making?, *European Heart Journal* **37**: 1196–1207.

7

Conclusions

In Chapters 1 and 2, the background and motivation for the study have been given and the mechanisms behind the heart pumping actions were explored. The need for a greater understanding of the behaviour of both a healthy and diseased (or damaged) heart was emphasised. This need formed the incentive to develop a computational efficient numerical analysis method which could be used to predict the twisting and change of volume of the left ventricle during a single heartbeat. The complexity of the multi-scale processes occuring from the molecular level up, to the contraction of the myocardium, was discussed. The review provided a necessary framework within which to place the simplified idealisations that followed in subsequent chapters. The decision to focus on the electro-mechanics of the left ventricle was made as a consequence of the primary importance of this chamber to any understanding of the pumping action of the heart. The blood flow dynamics was consider to be outside the scope of this research. This omission might appear rather odd, since the sole purpose of the heart is to move the blood. However, like many previous researchers (Göktepe and Kuhl; 2009) (Hurtado et al.; 2016) (Momenan; 2017), it was recognised that there is real value to be gained from taking care to reproduce the electro-mechanics first.

By way of introduction to the method for biomedical engineers not familiar with finite elements, the three dimensional FE stress analysis of a solid (using 8-noded hexahedral elements) was reviewed and code produced (based largely on the work

of others) in Chapter 3. The FE capability was extended from a linear analysis to geometrically non-linear analysis and hyperelastic material non-linearity introduced. As part of that work, a technique for handling near-incompressible solids was reviewed and coded, and a benchmark problem satisfactorily simulated. The MATLAB scripts for each (progressively more advanced) analysis were included and explained. In Chapter 4, the replacement of the hexahedral finite elements by shell elements was undertaken. The formulation of those elements and their coding was described in some detail. Challenging benchmark problems for the large deformation of the shell elements were successfully undertaken.

In Chapter 5, the numerical simulation of the electrophysiology was described and the Aliev-Panfilov model used to simulate the membrane potential behaviour in the myocardium. This ability was added to the Total Lagrangian shell FE formulation described in Chapter 4. The coupled analysis was then tested on a panel and the effects of changing both the time-step size and the mesh densities were explored. Both changes led to convergence of the solutions as finer resolutions were made. In Chapter 6, the importance of the helical shapes of the myocardial band was emphasised and a simplified Torrent-Guasp model developed by Grosberg was used to explore the effect of different parameters on the twist and ejection fraction produced by a beating heart.

It was apparent from the findings in Chapter 6 that changing the location where twisting is measured gave rise to significant differences. This finding echos the observations made by others on imaging techniques (Badano and Muraru; 2019). Therefore, a standardization of how and where the data should be processed is required so that direct comparisons can be made between simulations and imaging results.

Currently, there are no electro-mechanical coupled prediction models that are being used as a diagnostic tool by medical practitioners. Such models are currently under development. There are a great many aspects to consider and different research groups tend to focus on one of these feature; for example (i) the detail included in the ionic model (Lelli; 2012), (ii) the mechanical constitutive model (Holzapfel and Ogden; 2009), (iii) the ability to take advantage of multi-core computing hard-

ware (Ruth; 2014). The principle novelty of the work reported in this thesis is the development of a *Total Lagrangian* geometrically non-linear electro-mechanical coupled (degenerate) 9-noded shell-based *finite element* formulation used in conjunction with a simplified (Torrent-Guasp) model of the myocardial band representing part of the left and right ventricles. That band is excited through the Aliev-Panfilov ionic model. The geometry may seem incomplete, as it does not cover the whole heart structure, but this idealisation has shown its ability to capture the essence of the cardiac muscle pumping motion (Carreras et al.; 2016). Further improvement could be made by using a more sophisticated heart geometry (such as through the dataset available on PhysioNet) or modifications to the ionic model, to allow a smaller number of time steps through a cardiac cycle.

The open source *finite element* analysis code (Listing 5.1) has been written using MATLAB m-script. This code takes about 40 minutes to run an analysis using a single core of a 4 core 3.0GHz Intel i5-4590S CPU and 8 GB of DDR3 RAM (storing up to 28MB of variables), running Windows 7. All the script files and both the mechanical and the electrophysiology concepts used in the code have been included in this thesis. This enables others to extend the work further as part of the continuation of the journey to better understand the behaviour of the heart.

References

Badano, L. P. and Muraru, D. (2019). Editorial: Twist mechanics of the left ventricle, *Circulation: Cardiovascular Imaging* **12(4)**.

Carreras, F., Ballester, M., Pujadas, S., Leta, R. and Pons-Llado, G. (2016). Morphological and functional evidences of the helical heart from non-invasive cardiac imaging, *European Journal of Cardio-thoracic Surgery* **29S(S50-55)**.

Göktepe, S. and Kuhl, E. (2009). Computational modeling of cardiac electrophysiology: A novel finite element approach, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING* **79**: 156–178.

Holzapfel, G. A. and Ogden, R. W. (2009). Constitutive modelling of passive my-

ocardium: A structurally based framework for material characterization, *Philosophical Transactions of The Royal Society A* **367**: 3445–3475.

Hurtado, D. E., Castro, S. and Gizzi, A. (2016). Computational modeling of non-linear diffusion in cardiac electrophysiology: A novel porous-medium approach, *Computer Methods in Applied Mechanics and Engineering* **300(1)**: 70–83.

Lelli, C. (2012). *Characterization of Delayed After-Depolarization in Extended FitzHugh-Nagumo Models*, PhD thesis, Politecnico di Milano.

Momenan, B. (2017). *Development of a Thick Continuum-Based Shell Finite Element for Soft Tissue Dynamics*, PhD thesis, University of Ottawa.

Ruth, A. (2014). *Electromechanical Large Scale Computational Models of the Ventricular Myocardium*, PhD thesis, Universitat Politècnica de Catalunya.

References

Aggarwal, R., Mytton, O. T., Derbrew, M., Hananel, D., Heydenburg, M., Issenberg, B., MacAulay, C., Mancini, M. E., Morimoto, T., Soper, N., Ziv, A. and Reznick, R. (2010). Training and simulation for patient safety, *Quality and Safety in health Care* **19(2)**: 34–43.

Ahmad, S. (1969). *Curved Finite Elements in the Analysis of Solid, Shell and Plate Structures*, PhD thesis, University College of Swansea.

Aliev, R. R. and Panfilov, A. V. (1996). A simple two-variable model of cardiac excitation, *Chaos, Solitons & Fractals* **7**: 293–301.

Allan, C. K., Thiagarajan, R. R., Beke, D., Imprescia, A., Kappus, L. J., Garden, A., Hayes, G., Laussen, P. C., Bacha, E. and Weinstock, P. H. (2010). Simulation-based training delivered directy to the pediatric cardiac intensive care unit engenders preparedness, comofort and decreased anxiety among multidisciplinary resuscitation teams, *The Journal of Thoracic and Cardiovascular Surgery* **14(3)**: 646–652.

Ambrosi, D., Arioli, G., Nobile, F. and Quarteroni, A. (2011). Electromechanical coupling in cardiac dynamics: The active strain approach, *Society for Industrial and Applied Mathematics* pp. 605–621.

Ambrosi, D. and Pezzuto, S. (2011). Active stress vs. active strain in machanobiology: Constitutive issues, *Journal of Elasticity* **107**: 199–212.

Amr, A., Kayvanpour, E., Sedaghat-Hamedani, F., Passerini, T., Mihalef, V., Lai, A., Neumann, D., Georgescu, B., Buss, S., Mereles, D., Zitron, E., Posch, A. E., Würstle, M., Mansi, T., Katus, H. A. and Meder, B. (2016). Personalized computer simulation of diastolic function in heart failure, *Genomics Proteomics Bioinformatics* **14**: 244–252.

Badano, L. P. and Muraru, D. (2019). Editorial: Twist mechanics of the left ventricle, *Circulation: Cardiovascular Imaging* **12(4)**.

Baillargeon, B., Rebelo, N., Fox, D. D., Taylor, R. L. and Kuhl, E. (2014). The living heart project: A robust and integrative simulator for human heart function, *European Journal of Mechanics / A Solids* **48**: 38–47.

Bathe, K.-J. (2014). *Finite Element Procedures*, second edn, Prentice Hall, Pearson Education, Inc.

Bathe, K.-J. and Bolourchi, S. (1980). A geometric and material nonlinear plate and shell element, *Computers & Structures* **11**: 23–48.

Beeler, G. W. and Reuter, H. (1977). Reconstruction of the action potential of ventricular myocardial fibres, *The Journal Of Physiology* **268(1)**: 177–210.

Bonet, J. and Wood, R. D. (1997). *Nonlinear Continuum Mechanics for Finite Element Analysis*, Cambridge University Press.

British Heart Foundation (2003). Congenital heart disease statistics.

Buckberg, G. D., Nanda, N. C., Nguyen, C. and Kocica, M. J. (2018). What is the heart? anatomy, function, pathophysiology, and misconceptions, *Journal of Cardiovascular Development and Disease* **5(2)**.

Carreras, F., Ballester, M., Pujadas, S., Leta, R. and Pons-Llado, G. (2016). Morphological and functional evidences of the helical heart from non-invasive cardiac imaging, *European Journal of Cardio-thoracic Surgery* **29S(S50-55)**.

Coombs, W., Crouch, R. and Augarde, C. (2010). 70-line 3d finite deformation elastoplastic finite-element code, *Proceedings of the Seventh European Conference on Numerical Methods in Geotechnical Engineering* pp. 151–156.

Costabal, F. S., Concha, F. A., Hurtado, D. E. and Kuhl, E. (2017). The importance of mechano-electrical feedback and inertia in cardiac electromechanics, *Computer methods in applied mechanics and engineering* **320**: 352–368.

Dorri, F. (2004). *A Finite Element Model Of The Human Left Ventricular Systole, Taking Into Account The Fibre Orientation Pattern*, Swiss Federal Institute Of Technology Zurich.

Drake, R., Vogl, A. W. and Mitchell, A. (2009). *Gray's Anatomy for Students*, second edition edn, Churchill Livingstone.

Dr.S.Venkatesan (2008). What is the mechanism of pericardial rub ?, https://drsvenkatesan.com/2008/09/28/what-is-the-mechanism-of-pericardial-rub/.

ECG Pedia (2011). Av conduction, http://en.ecgpedia.org/index.php?title=AV_Conduction.

Employment, Social Policy, Health and Consumer Affairs (2004). 2586th council meeting, http://europa.eu/rapid/press-release_PRES-04-163_en.htm?locale=en.

Felippa, C. and Haugen, B. (2005). A unified formulation of small-strain corotational finite elements: I. theory, *Computer Methods in Applied Mechanics and Engineering* **194(21-24)**: 2285–2335.

Fitzhugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane, *Biophysical Journal* **1(6)**: 445–466.

Fujitsu (2015). Using supercomputers for safer heart surgery, http://www.fujitsu.com/global/vision/customerstories/university-of-tokyo/.

Göktepe, S., Acharya, S. N. S., Wong, J. and Kuhl, E. (2011). Computational modeling of passive myocardium, *International Journay for Numerical Methods in Biomedical Engineering* **27(1)**: 1–12.

Göktepe, S. and Kuhl, E. (2009). Computational modeling of cardiac electrophysiology: A novel finite element approach, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING* **79**: 156–178.

Göktepe, S. and Kuhl, E. (2010). Electromechanics of the heart: A unified approach to the strongly coupled excitationcontraction problem, *Computational Mechanics* **45(2)**: 227–243.

Gray, H. (2014). *Gray's Anatomy*, Churchill Livingstone Elsevier.

Grosberg, A. (2008). *A Bio-Inspired Computational Model of Cardiac Mechanics: Pathology and Development*, PhD thesis, California Institute of Technology.

Guyton, A. C. and Hall, J. E. (2011). *Textbook of Medical Physiology*, 12 edn, Saunders Elsevier.

Hall, J. E. (2011). *Guyton and Hall: Textbook of Medical Physiology*, twelfth edition edn, Saunders Elsevier.

Hodgkin, A. and Huxley, A. (1952). A quantitative description of membrane current and it's application to conduction and excitation in nerve, *The Journal of Physiology* **117**: 500–544.

Holzapfel, G. A. and Ogden, R. W. (2009). Constitutive modelling of passive myocardium: A structurally based framework for material characterization, *Philosophical Transactions of The Royal Society A* **367**: 3445–3475.

Hovels-Gurich, H. H., Konrad, K., Skorzenski, D., Nacken, C., Minkenberg, R., Messmer, B. J. and Seghaye, M.-C. (2006). Long-term neurodevelopmental outcome and exercise capacity after corrective surgery for tetralogy of fallot or ventricular septal defect in infancy, *The Annals Of Thoracic Surgery* **81**: 958–966.

Hurtado, D. E., Castro, S. and Gizzi, A. (2016). Computational modeling of nonlinear diffusion in cardiac electrophysiology: A novel porous-medium approach, *Computer Methods in Applied Mechanics and Engineering* **300(1)**: 70–83.

Jenkins, G. W., Kemnitz, C. P. and Tortora, G. J. (2010). *Anatomy and Physiology: From Science To Life*, 2 edn, John Wiley & Sons, Inc.

Jensen, L. R., Rauhe, J. C. and Stegmann, J. (2001). *Finite Elements for Geometric Non-Linear Analysis of Composite Laminates and Sandwich Structures*, PhD thesis, Aalborg University.

Kayvanpour, E., Mansi, T., Sedaghat-Hamedani, F., Amr, A., Neumann, D., Georgescu, B., Seegerer, P., Kamen, A., Hass, J., Frese, K. S., Irawati, M., Wirsz, E., King, V., Buss, S., Mereles, D., Zitron, E., Keller, A., Katus, H. A., Comaniciu, D. and Meder, B. (2015). Towards personalized cardiology: Multi-scale modeling of the failing heart, *PLoS ONE* **10(7)**.

Kerckhoffs, R. C., Healy, S. N., USYK, T. P. and McCulloch, A. D. (2006). Computational methods for cardiac electromechanics, *Proceedings of the IEEE* **94**: 769–783.

Kim, Y., Marom, E., Herndon, J. and McAdams, H. (2005). Pulmonary vein diameter, cross-sectional area and shape: Ct analysis, *Radiology* **235(1)**: 43–49.

Klabunde, R. E. (2008). Normal impulse conduction, `http://www.cvphysiology.com/Arrhythmias/A003.htm`.

Kotikanyadanam, M., Göktepe, S. and Kuhl, E. (2010). Computational modeling of electrocardiograms: A finite element approach toward cardiac excitation, *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING* **26**: 524–533.

Kuijpers, N. H. L., Hermeling, E., Bovendeerd, P. H. M., Delhaas, T. and Prinzen, F. W. (2012). Modeling cardiac electromechanics and mechanoelectrical coupling in dyssynchronous and failing hearts, *J. of Cardiovasc. Trans. Res.* **5**: 159–169.

Laulusa, A., Bauchau, O., Choi, J.-Y., Tan, V. and Li, L. (2006). Evaluation of some shear deformable shell elements, *International Journal of Solids and Structures* **43**: 5033–5054.

Lelli, C. (2012). *Characterization of Delayed After-Depolarization in Extended FitzHugh-Nagumo Models*, PhD thesis, Politecnico di Milano.

Levick, J. (2011). *An Introduction To Cardiovascular Physilogy*, fifth edition edn, Hodder Arnold.

Lewis, R. (1996). *The Finite Element Method in Heat Transfer Analysis*, John Wiley & Sons.

Luo, C. H. and Rudy, Y. (1991). A model of the ventricular cardiac action potential. depolarization, repolarization, and their interaction, *Circulation Research* **68(6)**: 1501–1526.

McGarvey, J. R., Mojsejenko, D., Dorsey, S. M., Nikou, A., Burdick, J. A., Gorman, J. H., Jackson, B. M., Pilla, J. J., Gorman, R. C. and Wenk, J. F. (2015). Temporal changes in infarct material properties: An in vivo assessment using magnetic

resonance imaging and finite element simulations, *The Society of Thoracic Surgeons* **100(2)**: 528–529.

McKinney, J., Cook, D. A., Wood, D. and Hatala, R. (2013). Simulation-based training for cardiac auscultation skills: Systematic review and meta-analysis, *Journal of General Internal Medicine* **28(2)**: 283–291.

Momenan, B. (2017). *Development of a Thick Continuum-Based Shell Finite Element for Soft Tissue Dynamics*, PhD thesis, University of Ottawa.

Nagumo, J., Arimoto, S. and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon, *Proceedings of the IRE* **50(10)**: 2061–2070.

Nash, M. P. and Panfilov, A. V. (2004). Electromechanical model of excitable tissue to study reentrant cardiac arrhythmias, *Progress in Biophysics & Molecular Biology* **85**: 501–522.

National Geographic (2013). How many cells are in your body?, `http://phenomena.nationalgeographic.com/2013/10/23/how-many-cells-are-in-your-body/`.

Nemavhola, F. (2019). Detailed structural assessment of healthy interventricular septum in the presence of remodeling infarct in the free wall - a finite element model, *Heliyon* **5(6)**.

Nichols, M., Townsend, N., Scarborough, P. and Rayner, M. (2014). Cardiovascular disease in europe 2014: Epidemiological update, *European Heart Journal* **35(42)**: 2950–2959.

Niederer, S. A., Kerfoot, E., Benson, A. P., Bernabeu, M. O., Bernus, O., Bradley, C., Cherry, E. M., Clayton, R., Fenton, F. H., Garny, A., Heidenreich, E., Land, S., Maleckar, M., Pathmanathan, P., Plank, G., Rodríguez, J. F., Roy, I., Sachse, F. B., Seemann, G., Skavhaug, O. and Smith, N. P. (2011). Verification of cardiac tissue electrophysiology simulators using an n-version benchmark, *Philosophical Transactions of The Royal Society* **369**: 4331–4351.

Niederer, S., Hunter, P. and Smith, N. (2006). A quantitative analysis of cardiac myocyte relaxation: A simulation study, *Biophysical Journal* **90**: 1697–1722.

Noble, D. (1962). A modification of the hodgkin - huxley equations applicable to purkinje fibre action and pacemaker potentials, *The Journal Of Physiology* **160(2)**: 317–352.

Ottosen, N. S. and Petersson, H. (1992). *Introduction to the Finite Element Method*, Prentice Hall International.

Parisch, H. (1995). A continuum-based shell theory for non-linear applications, *International Journal for Numerical Methods in Engineering* **38**: 1855–1883.

Plank, G., Burton, R. A., Hales, P., Bishop, M., Mansoori, T., Bernabeu, M. O., Garny, A., Prassal, A. J., Bollensdorff, C., Mason, F., Mahmood, F., Rodriguez, B., Grau, V., Schneider, J. E., Gavaghan, D. and Kohl, P. (2009). Generation of histo-anatomically representative models of the individual heart: tools and application, *Philosophical Transactions of The Royal Society A* **367**: 2257–2292.

Prince, M., Novelline, R., Athanasoulis, C. and Simon, M. (1983). The diameter of the inferior vena cava and its implication for the use of vena caval filters, *Radiology* **149(3)**: 687–689.

Rice, J. (2017). Introduction to finite element methods (asen 5007), `http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/`.

Roth, B. J. (1991). Action potential propagation in a thick strand of cardiac muscle, *Circulation Research* **68(1)**: 162–173.

Ruth, A. (2014). *Electromechanical Large Scale Computational Models of the Ventricular Myocardium*, PhD thesis, Universitat Politècnica de Catalunya.

Sallin, E. A. (1969). Fiber orientation and ejection fraction in the human left ventricle, *Biophysical Journal* **9(7)**: 954–964.

Scovazzi, G., Carnes, B. and Zeng, X. (2015). A simple, stable and accurate tetrahedral finite element for transient, nearly incompressible, linear and nonlinear elasticity: A dynamic variational multiscale approach, *International Journal for Numerical Methods in Engineering* **106(10)**.

Sengupta, P. P. and Narula, J. (2014). Cardiac strain as a universal biomarker: Interpreting the sounds of uneasy heart muscle cells, *JACC: Cardiovascular Imaging* **7(5)**: 534–536.

Sengupta, P. P., Tajik, J., Chandrasekaran, K. and Khandheria, B. K. (2008). Twist mechanics of the left ventricle: Principle and application, *JACC: Cardiovascular Imaging* **1(3)**: 366–376.

Sermesant, M., Chabiniok, R., Chinchapatnam, P., Mansi, T., Billet, F., Moireau, P., Peyrat, J., Wong, K., Relan, J., Rhode, K., Ginks, M., Lambiase, P., Delingette, H., Sorine, M., Rinaldi, C., Chapelle, D., Razavi, R. and Ayache, N. (2012). Patient-specific electromechanical models of the heart for the prediction of pacing acute effects in crt: A preliminary clinical validation, *Medical Image Analysis* **16**: 201–215.

Shuaiby, S. M., Hassan, M. A. and El-Melegy, M. (2011). Modeling and simulation of the action potential in human cardiac tissues using finite element method, *Journal of Communications and Computer Engineering* **2:3**: 21–27.

Simo, J. and Armero, F. (1992). Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes*, *International Journal for Numerical Methods in Engineering* **33**: 1413–1449.

Simoons, M. L. and Hugenholtz, P. (1975). Gradual changes of ecg waveform during and after exercise in normal subjects, *Circulation* **52(4)**: 570577.

Smiseth, O. A., Torp, H., Opdahl, A., Haugaa, K. H. and Urheim, S. (2016). Myocardial strain imaging: How useful is it in clinical decision making?, *European Heart Journal* **37**: 1196–1207.

Smith, N., de Vecchi, A., McCormick, M., Nordsletten, D., Camara, O., Frangi, A. F., Delingette, H., Sermesant, M., Relan, J., Ayache, N., Krueger, M. W., Schulze, W. H., Hose, R., Valverde, I., Beerbaum, P., Staicu, C., Siebes, M., Spaan, J., Hunter, P., Weese, J., Lehmann, H., Chapelle, D. and Rezavi, R. (2011). euheart: Personalized and integrated cardiac care using patient-specific cardiovascular modelling, *Interface Focus* **1**: 349–364.

Sugiura, S., Washio, T., Hatano, A., Okada, J., Watanabe, H. and Hisada, T. (2012). Multi-scale simulations of cardiac electrophysiology and mechanics using the university of tokyo heart simulator, *Progress in Biophysics and Molecular Biology* **110**: 380–389.

Sze, K., Liu, X. and Lo, S. (2004). Popular benchmark problems for geometric nonlinear analysis of shells, *Finite Elements in Analysis and Design* **40(11)**: 1551–1569.

Thomas, J. (1995). *Numerical Partial Differential Equations: Finite Difference Methods*, Springer-Verlag.

TOP500 (2016). K computer, sparc64 viiifx 2.0ghz, tofu interconnect, `https://www.top500.org/system/177232`.

Torrent-Guasp, F., Kocica, M. J., Corno, A. F., Komeda, M., Carreras-Costa, F., Flotats, A., Cosin-Aguillar, J. and Wen, H. (2005). Towards new understanding of the heart structure and function, *European Journal of Cardio-thoracic Surgery* **27**: 191–201.

Truong, Q. A., Massaro, J. M., Rogers, I. S., Mahabadi, A. A., Kriegel, M. F., Fox, C. S., O'Donnell, C. J. and Hoffmann, U. (2012). Reference values for normal pulmonary artery dimensions by noncontrast cardiac computed tomography, *Circulation: Cardiovascular Imaging* **5**: 147 – 154.

Wang, Y., Georgescu, B., Chen, T., Wu, W., Wang, P., Lu, X., Ionasec, R., Zheng, Y. and Comaniciu, D. (2013). Learning-based detection and tracking in medical imaging: A probabilistic approach, *Lecture Notes in Computational Vision and Biomechanics* **7**: 209–235.

Wolak, A., Gransar, H., Thomson, L. E., Friedman, J. D., Hachamovitch, R., Gustein, A., Shaw, L. J., Polk, D., Wong, N. D., Saouaf, R., Hayes, S. W., Rozanski, A., Slomka, P. J., Germano, G. and Berman, D. S. (2008). Aortic size assessment by noncontrast cardiac computed tomography: Normal limits by age, gender and body surface area, *The Journal of the American College of Cardiology* **1(2)**: 200–209.

Yamashita, T., Ueda, A., Mitsui, T., Tomonaga, A., Matsumoto, S., Kodama, T. and Fujitani, H. (2015). The feasibility of an efficient drug desgin method with high-performance computers, *Chemical and Pharmaceutical Bulletin* **63**: 147–155.

Zhang, Y., Bajaj, C. B. and Sohn, B.-S. (2005). 3d finite element meshing from imaging data, *Comput Methods Appl Mech Eng.* **194**: 48–49.

Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M. and Comaniciu, D. (2008). Four-chamber heart modeling and automatic segmentation for 3d cardiac ct volumes using marginal space learning and steerable features, *IEEE Transactions on Medical Imaging* **27(11)**: 1668–1681.

Appendix I

Common functions for the FE analyses

The recurring MATLAB function file that are is for all 8-noded hexahedral finite element analyses (Listings 3.3, 3.5 and 3.6) are given below.

```
function B=formB(dNx,nen)
B=zeros(6,nen*3); B(1,1:3:end)=dNx(1,:);
B(2,2:3:end)=dNx(2,:); B(3,3:3:end)=dNx(3,:);
B(4,1:3:end)=dNx(2,:); B(4,2:3:end)=dNx(1,:);
B(5,2:3:end)=dNx(3,:); B(5,3:3:end)=dNx(2,:);
B(6,1:3:end)=dNx(3,:); B(6,3:3:end)=dNx(1,:);

function [dNr]=dershapefunc2D(xsi,eta,zet)
ngp=size(xsi,1); r2=ngp*3;
dNr(1:2:r2,1)=-0.125*(1-eta).*(1-zet);
dNr(1:2:r2,2)=-0.125*(1-eta).*(1+zet);
dNr(1:2:r2,3)= 0.125*(1-eta).*(1+zet);
dNr(1:2:r2,4)= 0.125*(1-eta).*(1-zet);
dNr(1:2:r2,5)=-0.125*(1+eta).*(1-zet);
dNr(1:2:r2,6)=-0.125*(1+eta).*(1+zet);
dNr(1:2:r2,7)= 0.125*(1+eta).*(1+zet);
dNr(1:2:r2,8)= 0.125*(1+eta).*(1-zet);
dNr(2:2:r2,1)=-0.125*(1-xsi).*(1-zet);
dNr(2:2:r2,2)=-0.125*(1-xsi).*(1+zet);
dNr(2:2:r2,3)=-0.125*(1+xsi).*(1+zet);
dNr(2:2:r2,4)=-0.125*(1+xsi).*(1-zet);
dNr(2:2:r2,5)= 0.125*(1-xsi).*(1-zet);
dNr(2:2:r2,6)= 0.125*(1-xsi).*(1+zet);
dNr(2:2:r2,7)= 0.125*(1+xsi).*(1+zet);
dNr(2:2:r2,8)= 0.125*(1+xsi).*(1-zet);
dNr(3:2:r2,1)=-0.125*(1-xsi).*(1-eta);
dNr(3:2:r2,2)= 0.125*(1-xsi).*(1-eta);
dNr(3:2:r2,3)= 0.125*(1+xsi).*(1-eta);
dNr(3:2:r2,4)=-0.125*(1+xsi).*(1-eta);
dNr(3:2:r2,5)=-0.125*(1-xsi).*(1+eta);
dNr(3:2:r2,6)= 0.125*(1-xsi).*(1+eta);
dNr(3:2:r2,7)= 0.125*(1+xsi).*(1+eta);
```

```
33 dNr(3:2:r2,8)=-0.125*(1+xsi).*(1+eta);
```

Listing I.1: Function files required for the 8-noded hexahedral finite element analysis code.

For the 9-noded shell finite element analyses (Listings 4.2, II.4, II.5 and 5.2), the shared function files are given below.

```
function [B,GNL,epsE,L]=formBmatrix(N,dNr,zet,coord,dxr,...
                                    g1,g2,duvw,oL)
nen=size(coord,1);
invJ=inv(dxr);
dNx=dxr\[dNr; zeros(1,nen)];
G=zet*(invJ(:,1:2)*dNr)+invJ(:,3)*N;
B9=zeros(9,nen*5); BNL=zeros(6,nen*5); GNL=zeros(9,nen*5); L=zeros
   (1,9);
for n=1:3
  for m=1:nen
    L(:,(n-1)*3+1:(n-1)*3+3)=L(:,(n-1)*3+1:(n-1)*3+3)+...
     ([dNx(:,m) g1(m,n).*ones(3,1).*G(:,m)...
       g2(m,n).*ones(3,1).*G(:,m)]*...
      [duvw(((m-1)*5)+n); duvw(((m-1)*5)+4);
       duvw(((m-1)*5)+5)])';
  end
end
epsE=[0.5*(L(1,1)+L(1,1)+oL(1,1)*L(1,1)+...
      oL(1,4)*L(1,4)+oL(1,7)*L(1,7)+...
      L(1,1)*oL(1,1)+L(1,4)*oL(1,4)+L(1,7)*oL(1,7)+...
      L(1,1)*L(1,1)+L(1,4)*L(1,4)+L(1,7)*L(1,7));
      0.5*(L(1,5)+L(1,5)+oL(1,2)*L(1,2)+...
      oL(1,5)*L(1,5)+oL(1,8)*L(1,8)+...
      L(1,2)*oL(1,2)+L(1,5)*oL(1,5)+L(1,8)*oL(1,8)+...
      L(1,2)*L(1,2)+L(1,5)*L(1,5)+L(1,8)*L(1,8));
      0.5*(L(1,9)+L(1,9)+oL(1,3)*L(1,3)+...
      oL(1,6)*L(1,6)+oL(1,9)*L(1,9)+...
      L(1,3)*oL(1,3)+L(1,6)*oL(1,6)+L(1,9)*oL(1,9)+...
      L(1,3)*L(1,3)+L(1,6)*L(1,6)+L(1,9)*L(1,9));
      1.0*(L(1,2)+L(1,4)+oL(1,1)*L(1,2)+...
      oL(1,4)*L(1,5)+oL(1,7)*L(1,8)+...
      L(1,1)*oL(1,2)+L(1,4)*oL(1,5)+L(1,7)*oL(1,8)+...
      L(1,1)*L(1,2)+L(1,4)*L(1,5)+L(1,7)*L(1,8));
      1.0*(L(1,8)+L(1,6)+oL(1,3)*L(1,2)+...
      oL(1,6)*L(1,5)+oL(1,9)*L(1,8)+...
      L(1,3)*oL(1,2)+L(1,6)*oL(1,5)+L(1,9)*oL(1,8)+...
      L(1,3)*L(1,2)+L(1,6)*L(1,5)+L(1,9)*L(1,8));
      1.0*(L(1,3)+L(1,7)+oL(1,1)*L(1,3)+...
      oL(1,4)*L(1,6)+oL(1,7)*L(1,9)+...
      L(1,1)*oL(1,3)+L(1,4)*oL(1,6)+L(1,7)*oL(1,9)+...
      L(1,1)*L(1,3)+L(1,4)*L(1,6)+L(1,7)*L(1,9))];
B9([1 4 9],1:5:end)=dNx;
B9([5 2 6],2:5:end)=dNx;
B9([8 7 3],3:5:end)=dNx;
B9(:,      4:5:end)=g1(:,[1 2 3 1 2 2 3 3 1])'...
                    .*G([1 2 3 2 1 3 2 1 3],:);
B9(:,      5:5:end)=g2(:,[1 2 3 1 2 2 3 3 1])'...
                    .*G([1 2 3 2 1 3 2 1 3],:);
BL0=B9([1:3 5 7 9],:); BL0(4:6,:)=BL0(4:6,:)+B9([4 6 8],:);tL=oL+L;
A=[tL(:,1:3:9)  zeros(1,3)  zeros(1,3);
    zeros(1,3) tL(:,2:3:9)  zeros(1,3);
    zeros(1,3)  zeros(1,3) tL(:,3:3:9);
   tL(:,2:3:9) tL(:,1:3:9)  zeros(1,3);
    zeros(1,3) tL(:,3:3:9) tL(:,2:3:9);
   tL(:,3:3:9)  zeros(1,3) tL(:,1:3:9)];
for n=1:nen
  GNL(:,(n-1)*5+1:(n-1)*5+5)=...
    [dNx(1,n)*eye(3) g1(n,:)'.*G(1,n).*ones(3,1) g2(n,:)'...
     .*G(1,n).*ones(3,1);
     dNx(2,n)*eye(3) g1(n,:)'.*G(2,n).*ones(3,1) g2(n,:)'...
     .*G(2,n).*ones(3,1);
```

```
      dNx(3,n)*eye(3) g1(n,:)'.*G(3,n).*ones(3,1) g2(n,:)'...
      .*G(3,n).*ones(3,1)];
  BNL(:,(n-1)*5+1:(n-1)*5+5)=A*GNL(:,(n-1)*5+1:(n-1)*5+5);
end
B=BL0+BNL;

function [N]=shapefunc(xsi,eta)
N(:,1)= 0.25*xsi.*(xsi-1).*eta.*(eta-1);
N(:,2)=-0.50*xsi.*(xsi-1).*(eta+1).*(eta-1);
N(:,3)= 0.25*xsi.*(xsi-1).*eta.*(eta+1);
N(:,4)=-0.50*(xsi+1).*(xsi-1).*eta.*(eta+1);
N(:,5)= 0.25*xsi.*(xsi+1).*eta.*(eta+1);
N(:,6)=-0.50*xsi.*(xsi+1).*(eta+1).*(eta-1);
N(:,7)= 0.25*xsi.*(xsi+1).*eta.*(eta-1);
N(:,8)=-0.50*(xsi+1).*(xsi-1).*eta.*(eta-1);
N(:,9)=(xsi+1).*(xsi-1).*(eta+1).*(eta-1);

function [dNr]=dershapefunc2D(xsi,eta)
ngp=size(xsi,1); r2=ngp*2;
dNr(1:2:r2  ,1)= 1/4*eta.*(eta-1).*(2*xsi-1);
dNr(1:2:r2  ,2)=-1/2*(eta+1).*(eta-1).*(2*xsi-1);
dNr(1:2:r2  ,3)= 1/4*eta.*(eta+1).*(2*xsi-1);
dNr(1:2:r2  ,4)=-1*eta.*(eta+1).*xsi;
dNr(1:2:r2  ,5)= 1/4*eta.*(eta+1).*(2*xsi+1);
dNr(1:2:r2  ,6)=-1/2*(eta+1).*(eta-1).*(2*xsi+1);
dNr(1:2:r2  ,7)= 1/4*eta.*(eta-1).*(2*xsi+1);
dNr(1:2:r2  ,8)=-1*eta.*(eta-1).*xsi;
dNr(1:2:r2  ,9)= 2*(eta+1).*(eta-1).*xsi;
dNr(2:2:r2+1,1)= 1/4*xsi.*(xsi-1).*(2*eta-1);
dNr(2:2:r2+1,2)=-1*xsi.*(xsi-1).*eta;
dNr(2:2:r2+1,3)= 1/4*xsi.*(xsi-1).*(2*eta+1);
dNr(2:2:r2+1,4)=-1/2*(xsi+1).*(xsi-1).*(2*eta+1);
dNr(2:2:r2+1,5)= 1/4*xsi.*(xsi+1).*(2*eta+1);
dNr(2:2:r2+1,6)=-1*xsi.*(xsi+1).*eta;
dNr(2:2:r2+1,7)= 1/4*xsi.*(xsi+1).*(2*eta-1);
dNr(2:2:r2+1,8)=-1/2*(xsi+1).*(xsi-1).*(2*eta-1);
dNr(2:2:r2+1,9)= 2*(xsi+1).*(xsi-1).*eta;
```

Listing I.2: Function files required for the 9-noded shell finite element analysis code.

The function file which is shared by all the FE codes is given below

```
function [wp,GpLoc]=GpPos(ngp)
if ngp==8 %2x2x2
  wp=ones(8,1); g2=1/sqrt(3);
  xsi=[-1 -1 1 1 -1 -1 1 1].'*g2;
  eta=[-1 -1 -1 -1 1 1 1 1].'*g2;
  zet=[-1 1 1 -1 -1 1 1 -1].'*g2;
  GpLoc=[xsi eta zet];
elseif ngp==12 %2x2x3
  g2=1/sqrt(3); g3=sqrt(3/5); w1=8/9; w2=5/9;
  xsi=[-1 -1 -1  1  1  1 -1 -1 -1  1  1  1].'*g2;
  eta=[-1 -1 -1 -1 -1 -1  1  1  1  1  1  1].'*g2;
  zet=[-1  0  1 -1  0  1 -1  0  1 -1  0  1].'*g3;
  wp=[w2;w1;w2;w2;w1;w2;w2;w1;w2;w2;w1;w2];
  GpLoc=[xsi eta zet];
elseif ngp==18 %3x3x2
  g3=sqrt(3/5); g2=1/sqrt(3); w1=8/9; w2=5/9;
  xsi=  [-1;-1; 0; 0; 1; 1;-1;-1; 0; 0; 1; 1;-1;-1; 0; 0; 1; 1]*g3;
  eta=  [-1;-1;-1;-1;-1;-1; 0; 0; 0; 0; 0; 0; 1; 1; 1; 1; 1; 1]*g3;
  zet=  [-1; 1;-1; 1;-1; 1;-1; 1;-1; 1;-1; 1;-1; 1;-1; 1;-1; 1]*g2;
  w(:,1)=[w2;w2;w1;w1;w2;w2;w2;w2;w1;w1;w2;w2;w2;w2;w1;w1;w2;w2];
  w(:,2)=[w2;w2;w2;w2;w2;w2;w1;w1;w1;w1;w1;w1;w2;w2;w2;w2;w2;w2];
  wp=w(:,1).*w(:,2);
  GpLoc=[xsi eta zet];
```

```
elseif ngp==27 %3x3x3
  g3=sqrt(3/5); w1=8/9; w2=5/9;
  xsi=[-1; 0; 1;-1; 0; 1;-1; 0; 1; -1; 0; 1;-1; 0;
           1;-1; 0; 1; -1; 0; 1;-1; 0; 1;-1; 0; 1]*g3;
  eta=[-1;-1;-1;-1;-1;-1;-1;-1;-1;  0; 0; 0; 0; 0;
           0; 0; 0; 0;  1; 1; 1; 1; 1; 1; 1; 1; 1]*g3;
  zet=[-1;-1;-1; 0; 0; 0; 1; 1; 1; -1;-1;-1; 0; 0;
           0; 1; 1; 1; -1;-1;-1; 0; 0; 0; 1; 1; 1]*g3;
  w(:,1)=[w2;w1;w2;w2;w1;w2;w2;w1;w2; w2;w1;w2;w2;w1;w2;w2;w1;w2;
                                      w2;w1;w2;w2;w1;w2;w2;w1;w2];
  w(:,2)=[w2;w2;w2;w2;w2;w2;w2;w2;w2; w1;w1;w1;w1;w1;w1;w1;w1;w1;
                                      w2;w2;w2;w2;w2;w2;w2;w2;w2];
  w(:,3)=[w2;w2;w2;w1;w1;w1;w2;w2;w2; w2;w2;w2;w1;w1;w1;w2;w2;w2;
                                      w2;w2;w2;w1;w1;w1;w2;w2;w2];
  wp=w(:,1).*w(:,2).*w(:,3);
  GpLoc=[xsi eta zet];
end
```

Listing I.3: Function file required for all finite element analysis code.

Appendix II

Input function files for the FE analyses

The MATLAB input function file for the 8-noded hexahedral finite element cantilever analysis case is given below.

```
function [E,v,coord,etopol,bc,f]=cantilever_endload
E=1e9; v=0; nels=50; coord=zeros((nels-1)*4+8,3);
len=10; wid=0.1; thk=0.1; etopol=zeros(nels,8);
for nel=1:nels+1
  coord((nel-1)*4+1:(nel)*4,:)=[(nel-1)*(1/nels) 0 0;
                                (nel-1)*(1/nels) 0 thk;
                                (nel-1)*(1/nels) wid thk;
                                (nel-1)*(1/nels) wid 0];
  n1=(nel-1)*4+1; n2=n1+1; n3=nel*4+2; n4=n3-1;
  n5=n2+2; n6=n5-1; n7=n3+1; n8=n7+1;
  etopol(nel,:)=[n1 n2 n3 n4 n5 n6 n7 n8];
end
etopol(end,:)=[];
bc=[[1:12]' zeros(12,1)];
f=zeros(3*size(coord,1),1);
f(3*size(coord,1)-9:3:3*size(coord,1))=-0.25e+3;
```

Listing II.1: 16 line MATLAB script expressing the input data for the cantilever problem

The input file for the geometrically non-linear cantilever analysis with 8-noded hexahedral finite element is given below.

```
function [coord,etopol,fext,bc,D,ngp,lstps,NRitmax,NRtol]=...
                                        GNLcantilever_endload
E=1.2e6; v=0;
ngp=8; lstps=20; NRitmax=50; NRtol=1.0e-6;
nels=300; len=10; wid=1; thk=0.1;
Iz=zeros(6); Iz(1:3,1:3)=ones(3); I2=eye(6);
I2(4:6,4:6)=0.5*eye(3); D=zeros(6,6,ngp,nels);
etopol=zeros(nels,8);
for nel=1:nels+1
  coord((nel-1)*4+1:(nel)*4,:)=[(nel-1)*(len/nels) 0 0;
                                (nel-1)*(len/nels) 0 thk;
                                (nel-1)*(len/nels) wid thk;
                                (nel-1)*(len/nels) wid 0];
  n1=(nel-1)*4+1; n2=n1+1; n3=nel*4+2; n4=n3-1;
  n5=n2+2; n6=n5-1; n7=n3+1; n8=n7+1;
  etopol(nel,:)=[n1 n2 n3 n4 n5 n6 n7 n8];
  for gp=1:ngp
    D(:,:,gp,nel)=E/((1+v)*(1-2*v))*(v*Iz+((1-2*v)*I2));
  end
end
etopol(end,:)=[];
bc=[[1:12]' zeros(12,1)];
fext=zeros(3*size(coord,1),1);
fext(3*size(coord,1)-9:3:3*size(coord,1))=1;
```

Listing II.2: Input file for the geometrically non-linear cantilever subjected to an end load.

The input function file used for the incompressible Cook's membrane is given below.

```
function [coord,etopol,fext,bc,K,miu,ngp,lstps,NRitmax,NRtol]=...
  ICookMembrane
K=40.0942e4; miu=80.1938; ngp=8; lstps=10; NRitmax=50;
NRtol=1.0e-4; nelx=8; nely=8; etopol=[]; bc=[];
coord=zeros((nelx+1)*(nely+1)*2,3);
for elz=1:2
  for ely=1:nely+1
    for elx=1:nelx+1
      coord(elx+(ely-1)*(nelx+1)+(elz-1)*((nelx+1)*(nely+1)),:)=...
        [(elx-1)*48/nelx (elx-1)*(44/nelx)+((nelx+1-elx)*(((ely-1)
   *...
         ((44/nely)-(16/nely)))/nelx))+(ely-1)*(16/nely) elz-1];
    end
  end
end
%etopol
for ely=1:nely
  for elx=1:nelx
    etopol=[etopol; elx+(ely-1)*(nelx+1)...
                    elx+(ely-1)*(nelx+1)+((nelx+1)*(nely+1))...
                    elx+1+(ely-1)*(nelx+1)+((nelx+1)*(nely+1))...
                    elx+1+(ely-1)*(nelx+1)...
                    elx+(ely*(nelx+1))...
                    elx+(ely*(nelx+1))+((nelx+1)*(nely+1))...
                    elx+1+(ely*(nelx+1))+((nelx+1)*(nely+1))...
                    elx+1+(ely*(nelx+1))];
  end
end
%bc
for n=1:size(coord,1)
  if coord(n,1)==0
    bc=[bc; n*3-2 0; n*3-1 0];
  end
  bc=[bc; n*3 0];
end
%fext
load=100/(4*nely);
fext=zeros(size(coord,1)*3,1);
for n=1:size(coord,1)
  if coord(n,1)==48
    if coord(n,2)==44 || coord(n,2)==60
      fext(n*3-1)=load;
    else
      fext(n*3-1)=2*load;
    end
  end
end
```

Listing II.3: Input files for the incompressible Cook's membrane problem.

The input function file used for the shell cantilever that is subjected to a transverse end load is given below.

```
function [coord,etpl,fext,bc,ngp,lstps,NRitmax,NRtol,...
          ndim,D,dxr]=C_EL
ngp=8; NRitmax=50; NRtol=1e-6; ndim=5; lstps=20;
nels=8; len=10; dep=0.1; wid=1; lay=1;
E=1.2e6; nu=0; k=5/6; nen=9;
E=E*ones(1,lay); oVn=zeros(nen,3,nels);
coord=zeros((nels*2+1)*3,3+lay); D=zeros(6,6,ngp,lay,nels);
etpl=zeros(nels,nen); dxr=zeros(3,3,ngp,lay,nels);
coord(:,4:4+lay-1)=dep/lay;
for i=1:(nels*2+1)
  coord((i-1)*3+1,1:2)=[(i-1)*len/(2*nels) 0];
  coord((i-1)*3+2,1:2)=[(i-1)*len/(2*nels) wid/2];
  coord((i-1)*3+3,1:2)=[(i-1)*len/(2*nels) wid];
end
for i=1:nels
  n1=(i-1)*6+1; n2=n1+1; n3=n1+2; n4=n1+5; n5=n1+8;
  n6=n1+7; n7=n1+6; n8=n1+3; n9=n1+4;
  etpl(i,:)=[n1 n2 n3 n4 n5 n6 n7 n8 n9];
  oVn(:,:,i)=[zeros(nen,2) ones(nen,1)];
end
nodes=size(coord,1);
fext=zeros(nodes*ndim,1);
fext( nodes*5   -2)=04*1/6;
fext((nodes-1)*5-2)=04*2/3;
fext((nodes-2)*5-2)=04*1/6;
bc=zeros(nodes*5,2); n=0;
for i=1:nodes
  if coord(i,1)==0
    bc(i*5-4,:)=[i*5-4 0]; n=n+1;
    bc(i*5-3,:)=[i*5-3 0]; n=n+1;
    bc(i*5-2,:)=[i*5-2 0]; n=n+1;
    bc(i*5-1,:)=[i*5-1 0]; n=n+1;
    bc(i*5  ,:)=[i*5   0]; n=n+1;
  end
end
bc=sortrows(bc,1);
bc(1:nodes*5-n,:)=[];

ex=[1 0 0].'; ey=[0 1 0].'; ez=[0 0 1].';
for nel=1:nels
  elcoord=coord(etpl(nel,:),1:3);
  t=coord(etpl(nel,:),4:4+lay-1); T=sum(t,2);
  [~,GpLoc]=GpPos(ngp);
  for n=1:lay
    Dn=(E(n)/(1-nu^2))*[[1 nu; nu 1; 0 0] zeros(3,4);...
      zeros(3,3) eye(3)*k*(1-nu)*0.5];
    Dn(4,4)=Dn(4,4)/k;
    for gp=1:ngp
      xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zeta=GpLoc(gp,3);
      N  =shapefunc(xsi,eta);
      zet=-1+(2*sum(N*t(:,1:n))-N*(t(:,n).*(1-zeta)))/(N*T);
      dNr=dershapefunc2D(xsi,eta);
      dxr(1:2,:,gp,n,nel)=(dNr*(elcoord+0.5*zet*(oVn(:,:,nel)...
                          .*(T*ones(1,3)))));
      dxr(3,:,gp,n,nel)=(0.5*(N.*T')*oVn(:,:,nel));
      VnGp=(N*oVn(:,:,nel)).';
      sdir=dxr(2,:,gp,n,nel)'/norm(dxr(2,:,gp,n,nel)');
      er=cross(sdir,VnGp); er=er/norm(er);
      es=cross(VnGp,er);   es=es/norm(es);
      et=VnGp/norm(VnGp);
      l1=(ex.'*er); m1=(ey.'*er); n1=(ez.'*er);
      l2=(ex.'*es); m2=(ey.'*es); n2=(ez.'*es);
```

```
      l3=(ex.'*et); m3=(ey.'*et); n3=(ez.'*et);
      Q=[       l1*l1        m1*m1        n1*n1...
                l1*m1        m1*n1        n1*l1 ;
                l2*l2        m2*m2        n2*n2...
                l2*m2        m2*n2        n2*l2 ;
                l3*l3        m3*m3        n3*n3...
                l3*m3        m3*n3        n3*l3 ;
             2*l1*l2      2*m1*m2      2*n1*n2...
          l1*m2+l2*m1 m1*n2+m2*n1 n1*l2+n2*l1 ;
             2*l2*l3      2*m2*m3      2*n2*n3...
          l2*m3+l3*m2 m2*n3+m3*n2 n2*l3+n3*l2 ;
             2*l3*l1      2*m3*m1      2*n3*n1...
          l3*m1+l1*m3 m3*n1+m1*n3 n3*l1+n1*l3];
      D(:,:,gp,n,nel)=Q'*Dn*Q;
    end
  end
end
```

Listing II.4: Input file for the geometrically non-linear cantilever subjected to an end load.

The input function file used for the shell cantilever that is subjected to a end moment is given below.

```
function [coord,etpl,fext,bc,ngp,lstps,NRitmax,NRtol,...
          ndim,D,dxr]=C_M
ngp=8; NRitmax=50; NRtol=1e-6; ndim=5; lstps=60;
nels=8; len=12; dep=0.1; wid=1; lay=1;
E=1.2e6; nu=0; k=5/6; nen=9;
E=E*ones(1,lay); oVn=zeros(nen,3,nels);
coord=zeros((nels*2+1)*3,3+lay); D=zeros(6,6,ngp,lay,nels);
etpl=zeros(nels,nen); dxr=zeros(3,3,ngp,lay,nels);
coord(:,4:4+lay-1)=dep/lay;
for i=1:(nels*2+1)
  coord((i-1)*3+1,1:2)=[(i-1)*len/(2*nels) 0];
  coord((i-1)*3+2,1:2)=[(i-1)*len/(2*nels) wid/2];
  coord((i-1)*3+3,1:2)=[(i-1)*len/(2*nels) wid];
end
for i=1:nels
  n1=(i-1)*6+1; n2=n1+1; n3=n1+2; n4=n1+5; n5=n1+8;
  n6=n1+7; n7=n1+6; n8=n1+3; n9=n1+4;
  etpl(i,:)=[n1 n2 n3 n4 n5 n6 n7 n8 n9];
  oVn(:,:,i)=[zeros(nen,2) ones(nen,1)];
end
nodes=size(coord,1);
fext=zeros(nodes*ndim,1);
fext( nodes*5   -0)=-25*pi/3*1/6;
fext((nodes-1)*5-0)=-25*pi/3*2/3;
fext((nodes-2)*5-0)=-25*pi/3*1/6;
bc=zeros(nodes*5,2); n=0;
for i=1:nodes
  if coord(i,1)==0
    bc(i*5-4,:)=[i*5-4 0]; n=n+1;
    bc(i*5-3,:)=[i*5-3 0]; n=n+1;
    bc(i*5-2,:)=[i*5-2 0]; n=n+1;
    bc(i*5-1,:)=[i*5-1 0]; n=n+1;
    bc(i*5  ,:)=[i*5   0]; n=n+1;
  end
end
bc=sortrows(bc,1);
bc(1:nodes*5-n,:)=[];

ex=[1 0 0].'; ey=[0 1 0].'; ez=[0 0 1].';
for nel=1:nels
  elcoord=coord(etpl(nel,:),1:3);
  t=coord(etpl(nel,:),4:4+lay-1); T=sum(t,2);
  [~,GpLoc]=GpPos(ngp);
  for n=1:lay
    Dn=(E(n)/(1-nu^2))*[[1 nu; nu 1; 0 0] zeros(3,4);...
      zeros(3,3) eye(3)*k*(1-nu)*0.5];
    Dn(4,4)=Dn(4,4)/k;
    for gp=1:ngp
      xsi=GpLoc(gp,1); eta=GpLoc(gp,2); zeta=GpLoc(gp,3);
      N  =shapefunc(xsi,eta);
      zet=-1+(2*sum(N*t(:,1:n))-N*(t(:,n).*(1-zeta)))/(N*T);
      dNr=dershapefunc2D(xsi,eta);
      dxr(1:2,:,gp,n,nel)=(dNr*(elcoord+0.5*zet*(oVn(:,:,nel)...
                          .*(T*ones(1,3)))));
      dxr(3,:,gp,n,nel)=(0.5*(N.*T')*oVn(:,:,nel));
      VnGp=(N*oVn(:,:,nel)).';
      sdir=dxr(2,:,gp,n,nel)'/norm(dxr(2,:,gp,n,nel)');
      er=cross(sdir,VnGp); er=er/norm(er);
      es=cross(VnGp,er);   es=es/norm(es);
      et=VnGp/norm(VnGp);
      l1=(ex.'*er); m1=(ey.'*er); n1=(ez.'*er);
      l2=(ex.'*es); m2=(ey.'*es); n2=(ez.'*es);
```

```
      l3=(ex.'*et); m3=(ey.'*et); n3=(ez.'*et);
      Q=[       l1*l1        m1*m1        n1*n1...
                l1*m1        m1*n1        n1*l1 ;
                l2*l2        m2*m2        n2*n2...
                l2*m2        m2*n2        n2*l2 ;
                l3*l3        m3*m3        n3*n3...
                l3*m3        m3*n3        n3*l3 ;
             2*l1*l2      2*m1*m2      2*n1*n2...
          l1*m2+l2*m1 m1*n2+m2*n1 n1*l2+n2*l1 ;
             2*l2*l3      2*m2*m3      2*n2*n3...
          l2*m3+l3*m2 m2*n3+m3*n2 n2*l3+n3*l2 ;
             2*l3*l1      2*m3*m1      2*n3*n1...
          l3*m1+l1*m3 m3*n1+m1*n3 n3*l1+n1*l3];
      D(:,:,gp,n,nel)=Q'*Dn*Q;
    end
  end
end
```

Listing II.5: Input file for the geometrically non-linear cantilever subjected to an end load.

Appendix III

Ejection fraction and twist calculation functions

```
function [EF, rots]=EjectionFraction(coord,etopol,uvw)
nodes=size(coord,1); coord=coord(:,1:3); elems=size(etopol,1);
uvw=reshape(uvw,5,nodes)'; uvw=uvw(:,1:3); mcoord=coord;
koord=coord+(1*uvw); kmcoord=koord;
oVn=zeros(9,3,elems); Vn=oVn;
TotalVolume0=0; TotalVolume1=0; oT=0.5/100; T=oT*1;
minrot=0.0; maxrot=0.0; nodes=size(coord,1); rots=zeros(nodes,2);
for node=1:nodes
  deltheta=atan2(koord(node,2),koord(node,1))...
           -atan2(coord(node,2),coord(node,1));
  if deltheta>pi
    deltheta=pi-deltheta;
  elseif deltheta<-2*pi
    deltheta=deltheta+2*pi;
  end
  rots(node,1:2)=[node deltheta*180/pi];
end

circelements=64; xielements=2; elem=0;
for xielement=1:xielements
  for circelement=1:circelements
    elem=elem+1;
    xsi=[-1; -1; -1; 0; 1; 1; 1; 0; 0];
    eta=[-1; 0; 1; 1; 1; 0; -1; -1; 0];
    dNr=dershapefunc2D(xsi,eta);
    dnr=dNr(1:2:end,:); dns=dNr(2:2:end,:);
    for n=1:9
      oVn(n,:,elem)=cross((dnr(n,:)*mcoord(etopol(elem,:),:))...
                    /norm(dnr(n,:)*mcoord(etopol(elem,:),:)),...
                          (dns(n,:)*mcoord(etopol(elem,:),:))...
                    /norm(dns(n,:)*mcoord(etopol(elem,:),:)));
      Vn(n,:,elem)=cross((dnr(n,:)*kmcoord(etopol(elem,:),:))...
                   /norm(dnr(n,:)*kmcoord(etopol(elem,:),:)),...
                         (dns(n,:)*kmcoord(etopol(elem,:),:))...
                   /norm(dns(n,:)*kmcoord(etopol(elem,:),:)));
    end
    coord(etopol(elem,:),:)=mcoord(etopol(elem,:),:)...
                            -0.5*oVn(:,:,elem).*(oT*ones(1,3));
    koord(etopol(elem,:),:)=kmcoord(etopol(elem,:),:)...
```

```
                              -0.5*Vn(:,:,elem).*(T*ones(1,3));
    n1=etopol(elem,1); n2=etopol(elem,2); n3=etopol(elem,3);
    n4=etopol(elem,4); n5=etopol(elem,5); n6=etopol(elem,6);
    n7=etopol(elem,7); n8=etopol(elem,8); n9=etopol(elem,9);
    if (circelement<0.5*circelements)
      for quad=1:4
        etopolquads=[n2 n3 n4 n9; n9 n4 n5 n6 ;
                     n8 n9 n6 n7; n1 n2 n9 n8];
        coordupper=[0 0 (max([coord(etopolquads(quad,1),3)...
                              coord(etopolquads(quad,2),3)...
                              coord(etopolquads(quad,3),3)...
                              coord(etopolquads(quad,4),3)]))];
        coordlower=[0 0 (min([coord(etopolquads(quad,1),3)...
                              coord(etopolquads(quad,2),3)...
                              coord(etopolquads(quad,3),3)...
                              coord(etopolquads(quad,4),3)]))];
        T1=[coordlower; coordupper;
            coord(etopolquads(quad,2),:) ;
            coord(etopolquads(quad,1),:)];
        VolT(1)=VolumeTet(T1);
        T2=[coordlower; coord(etopolquads(quad,1),:) ;
                        coord(etopolquads(quad,2),:) ;
                        coord(etopolquads(quad,3),:)];
        VolT(2)=VolumeTet(T2);
        T3=[coordlower; coord(etopolquads(quad,1),:) ;
                        coord(etopolquads(quad,3),:) ;
                        coord(etopolquads(quad,4),:)];
        VolT(3)=VolumeTet(T3);
        Volwedge=VolT(1)+VolT(2)+VolT(3);
        TotalVolume0=TotalVolume0+Volwedge;

        koordupper=[0 0 (max([koord(etopolquads(quad,1),3)...
                              koord(etopolquads(quad,2),3)...
                              koord(etopolquads(quad,3),3)...
                              koord(etopolquads(quad,4),3)]))];
        koordlower=[0 0 (min([koord(etopolquads(quad,1),3)...
                              koord(etopolquads(quad,2),3)...
                              koord(etopolquads(quad,3),3)...
                              koord(etopolquads(quad,4),3)]))];
        T1n=[koordlower; koordupper; koord(etopolquads(quad,2),:) ;
                                     koord(etopolquads(quad,1),:)];
        VolTn(1)=VolumeTet(T1n);
        T2n=[koordlower; koord(etopolquads(quad,1),:) ;
                         koord(etopolquads(quad,2),:) ;
                         koord(etopolquads(quad,3),:)];
        VolTn(2)=VolumeTet(T2n);
        T3n=[koordlower; koord(etopolquads(quad,1),:) ;
                         koord(etopolquads(quad,3),:) ;
                         koord(etopolquads(quad,4),:)];
        VolTn(3)=VolumeTet(T3n);
        Volwedgen=VolTn(1)+VolTn(2)+VolTn(3);
        TotalVolume1=TotalVolume1+Volwedgen;
      end
    end
  end
end
EF=100*(TotalVolume0-TotalVolume1)/TotalVolume0;

function VolT=VolumeTet(TetNodalCoords)
a=TetNodalCoords(1,:)'; b=TetNodalCoords(2,:)';
c=TetNodalCoords(3,:)'; d=TetNodalCoords(4,:)';
VolT=abs((a-d)'*(cross((b-d)',(c-d)'))')/6;
```

Listing III.1: Function files used for the computation of the ejection fraction and twist.

Appendix IV

Helical cylinder mesh generation function

To produce the helical cylindrical meshes seen in Figure 6.2, the following function was used to replace the early part of Listing 5.4.

```
function [coord, etpl, Vn, thk, bc]=spiralcyl()
coord=[];
hwidth=0.030;     % band half 'height' in metres
thk=0.005;        % radial thickness in metres
radius=0.05;      % radius in centimetres
turns=5;
circelements=72; xielements=2; thetaend=turns*2*pi;
helixAngdeg=70;  % helix angle
hfactor=pi*radius/(hwidth*tan(helixAngdeg*pi/180));

zeta=0; xikount=0;
for xi=+1:-1/xielements:-1
  xikount=xikount+1;
  k=0;
  for theta=0:thetaend/(2*circelements):thetaend
    xiwidth=xi*hwidth;
    k=k+1;
    thetas(k)=theta;
    zcent=10-(1.01*hfactor*theta*hwidth/pi);
    z(k,xikount)=zcent+xiwidth;
    r(k,xikount)=radius+zeta;
    y(k,xikount)=r(k,xikount)*sin(thetas(k));
    x(k,xikount)=r(k,xikount)*cos(thetas(k));
    coord=[coord; [x(k,xikount) y(k,xikount) z(k,xikount)]];
  end
end

etpl=[];
for xielement=1:xielements
  for circelement=1:circelements
    n1=((xielement-1)*2*((2*circelements)+1))+(2*(circelement-1))
   +1;
    n2=n1+1;
    n3=n2+1;
    n4=n3+(2*circelements)+1;
```

```
    n5=n4+(2*circelements)+1;
    n6=n5-1;
    n7=n6-1;
    n8=n4-2;
    n9=n8+1;
    elemnodes=[n1 n2 n3 n4 n5 n6 n7 n8 n9]; etpl=[etpl; elemnodes];
  end
end
bc=[];
for n=1:3
  bc=[bc; n*5-4 0; n*5-3 0; n*5-2 0];
end
for n=723:725
  bc=[bc; n*5-4 0; n*5-3 0; n*5-2 0];
end

nels=size(etpl,1); Vn=zeros(9,3,nels);
for nel=1:nels
  elcoord=coord(etpl(nel,:),:);
  xsi=[-1; -1; -1; 0; 1; 1; 1; 0; 0]; eta=[-1; 0; 1; 1; 1; 0; -1;
   -1; 0];
  dNr=dershapefunc2D(xsi,eta,9);
  dnr=dNr(1:2:end,:); dns=dNr(2:2:end,:);
  for n=1:9
    Vn(n,:,nel)=cross((dnr(n,:)*elcoord)/norm(dnr(n,:)*elcoord),(
   dns(n,:)*elcoord)/norm(dns(n,:)*elcoord));
  end
end
```

Listing IV.1: The function used to create the input for Listing 5.4 to enable the helical cylinders to be analysed.

Appendix V

Simplified Torrent-Guasp mesh generation function

To produce the simplified Torrent-Guasp meshes used in Chapter 6, the following function was used to replace the early part of Listing 5.4.

```
function [coord, etpl, Vn, thk, bc]=spiralheart2()
coord=[]; rFactor=1.0;
C0=4.9; C1=3.5; C2=1.3; C3=2; C4=1/6; C5=1; C6=1; C7=0.847;
width=1.25;     % band 'height' in centimetres
thk=0.5;        % radial thickness in centimetres
thetacrossing=pi*(-1+cot(pi/(C2*C3))); thetaright=3.8*pi;
circelements=2*16; xielements=2; thetaend=4.00*pi;
zeta=thk/2; xikount=0;
for xi=+1:-2/(2*xielements):-1
  xikount=xikount+1;
  k=0;
  for theta=0:thetaend/(2*circelements):thetaend
    widening=(pi/6)+(atan(theta-(2*pi)))/3; widening=0;
    xiwidth=xi*(width*(1+widening)/2);
    k=k+1;
    thetas(k)=theta;
    zcent=C0-(C1*thetas(k)^C2)/exp(thetas(k)/C3);
    Factor1=1+C4*cos((thetas(k)-thetacrossing)/2);
    Factor2=1+(C5/exp(((thetas(k)-thetaright)^2)));
    z(k,xikount)=zcent+xiwidth;
    discrim=(z(k,xikount)/C6)-C7;
    r(k,xikount)=rFactor*Factor1*Factor2*sqrt(discrim);
    y(k,xikount)=r(k,xikount)*sin(thetas(k));
    x(k,xikount)=r(k,xikount)*cos(thetas(k));
    coord=[coord; [x(k,xikount) y(k,xikount) z(k,xikount)]];
  end
end
coord=(1e-2)*coord; thk=(1e-2)*thk;

etpl=[];
for xielement=1:xielements
  for circelement=1:circelements
    n1=((xielement-1)*2*((2*circelements)+1))+...
                          (2*(circelement-1))+1;
```

```
    n2=n1+1;
    n3=n2+1;
    n4=n3+(2*circelements)+1;
    n5=n4+(2*circelements)+1;
    n6=n5-1;
    n7=n6-1;
    n8=n4-2;
    n9=n8+1;
    etpl=[etpl; [n1 n2 n3 n4 n5 n6 n7 n8 n9]];
  end
end

bcc=1;
for bcn=1:2*xielements
  bcc=[bcc; bcn*((2*circelements)+1); bcn*((2*circelements)+1)+1];
end
bcc=[bcc; (bcn+1)*((2*circelements)+1)];

bc=[];
for bcn=1:size(bcc,1)
  bc=[bc; bcc(bcn)*5-4 0; bcc(bcn)*5-3 0];
end

[~, apex]=min(coord(:,3));
bc=[bc; apex*5-2 0];

nels=size(etpl,1); Vn=zeros(9,3,nels);
for nel=1:nels
  elcoord=coord(etpl(nel,:),:);
  xsi=[-1; -1; -1; 0; 1; 1; 1; 0; 0];
  eta=[-1; 0; 1; 1; 1; 0; -1; -1; 0];
  dNr=dershapefunc2D(xsi,eta);
  dnr=dNr(1:2:end,:); dns=dNr(2:2:end,:);
  for n=1:9
    Vn(n,:,nel)=cross((dnr(n,:)*elcoord)/norm(dnr(n,:)*elcoord),...
                      (dns(n,:)*elcoord)/norm(dns(n,:)*elcoord));
  end
end
```

Listing V.1: The function used to create the input for Listing 5.4 to enable the simplified Torrent-Guasp myocardial band to be analysed.